Bound by
ABBEY BOOKBINDING & PRINTING
Unit 3 Gabalfa Workshops Excelsior Ind. Est. Cardiff CF14 3AY
Tel: (029) 2062 3290 Fax: (029) 2062 5420
Email: info@abbeybookbinding.co.uk
Web: www.abbeybookbinding.co.uk

NON-INTRUSIVE LOAD MONITORING WITH CANOPY CLUSTERING

Daniel Carr

A submission presented in partial fulfilment of the

requirements of the University of Glamorgan/Prifysgol Morgannwg

for the degree of Doctor of Philosophy

This research programme was carried out

in collaboration with Kigg Ltd.

September 2012

Table of Contents

Table of Figures

Abstract

Dwindling fossil fuels and the rising price of energy has meant that attitudes towards energy usage have changed in both domestic and commercial settings. This change in attitude has led to the development of smart metering technologies that are currently being rolled out across the world.

The research has been developed to be able to add functionality to smart metering devices by providing information about energy usage within the premises through Non-Intrusive Load Monitoring (NILM). The thesis provides a detailed description of the work undertaken to develop a novel method of load disaggregation within NILM to aid in the monitoring of energy usage and the provision of consumer feedback which can be integrated into smart metering technologies.

The research aims to provide a novel approach to NILM through the use of canopy clustering for its main process of load disaggregation. Canopy clustering provides the necessary tools for separating out appliances and groups of appliances for later classification into individual loads, which brings many benefits compared to other technologies.

The research methodology has been developed with robust techniques of data gathering, model development and validation through a rigorous testing approach. Real world examples of loads have been used for the creation and development of the models. The use of contemporary appliances within the research has meant that the NILM algorithm developed is current and usable. In the final implementation it could be commercialised for use by the general public.

The full procedures of the algorithm have been explained in detail with the addition of information on the final classification methods that could be used when implemented within smart metering devices. Further work and improvements to the research have also been included for consideration.

Glossary

NILM – Non-Intrusive Load Monitoring

EPSRC – Engineering and Physical Science Research Council

ISO – International Organisation for Standardisation

CAPEM^3 – Carbon Profiling through energy Measurement Metering and Modelling

P – Active Power

Q – Reactive Power

Var – Volt Ampere Reactive

W – Watts

kWh – Kilowatt Hour

CO_2 – Carbon Dioxide

AMR – Automatic Meter Reading

AMI – Advanced Metering Infrastructure

USP – Unique Selling Point

I_{out} – Current Output

I_{in} – Current Input

A – Ampere

V – Voltage

R – Resistance

RMS – Root Mean Square

N – Number of Turns

μ_0 – Permeability of Air

B – Magnetic Flux

CSV – Comma Separated Variable

KCL – Kirchhoff's Current Law

HVAC – Heating, Ventilation and Air Conditioning

ADC – Analogue to Digital Convertor

Hz – Hertz

FFT – Fast Fourier Transform

DFT – Discrete Fourier Transform

DSP – Digital Signal Processing

PCA – Principle Component Analysis

LDA – Linear Discriminant Analysis

Chapter 1. Objectives and Motivation

1.1. Background

The research commenced in 2008 on a project sponsored by the Engineering and Physical Science Research Council (EPSRC) in conjunction with Kigg Ltd. KIGG is a leading British ISO 9001:2008 certified company engaged in the design, development, manufacture and supply of kWh/electricity meters and greenhouse gas analysers. The initial project title for the research was 'CArbon Usage Profiling through Energy Measurement Metering and Modelling' ($CAPEM^3$), which was to consider the relationships between the fuel mix used within the electricity producing industry, and the periods that different appliances were being used.

The initial decisions behind the project were to be able to inform energy consumers how their actions were affecting the environment by providing feedback in terms of carbon production due to the usage of the electricity. This required detailed information of the makeup of the fuels used within the distribution network at time of use, which could then be mapped to the individuals' energy usage and converted to tonnes of Carbon Dioxide (CO_2) emissions for the time of use.

By providing energy users with information such as carbon usage, the ability to change people's behaviour could be realised, and steps could be taken to participate in load shifting during peak times to reduce the requirement for high carbon output power stations to be used. One of the main issues with this was that providing feedback to consumers in terms of Carbon does not give the whole picture and could confuse the user as to how they could change their behaviour to reduce their Carbon footprint.

The research programme naturally moved towards the direction of providing customer feedback as to how and where energy was being consumed within the premises. This was one of the initial drivers for researching methods used for

disaggregating loads at the electricity point of entry to the premises, and the concept of Non-Intrusive Load Monitoring (NILM) was investigated.

Early research into NILM was developed around the ideas of classification of large electrical goods within the domestic environment, and as research methods developed over time, the classification processes became more complex in their implementation with the movement from just steady state changes within the power signal being monitored to the use of harmonic analysis of the supply current waveform.

The uses of harmonic analysis become an integral part of the research, with the classification of the different harmonics being considered in greater depth. Investigating the frequency components within the energy supply were seen as a method of identifying the loads that were being consumed within the premises. For profiles of the loads to be constructed, clustering techniques were used to group like for like data points within the harmonic domains, which was then used for load classification.

The use of canopy clustering was investigated as a method of clustering the data to create the profiles. The canopy clustering algorithm was preferred over different clustering techniques due to its ability to be applied to the data set without any prior knowledge of the amount of clusters that would require identification, and therefore satisfied one of the design requirements of the research.

1.2. Motivation

The motivation behind the research was the development of a system that could be used to change the way in which consumers' use and view energy consumption. By providing feedback to the end-users of energy, they would be able to make informed decisions about how and where to make changes to their usage behaviour.

With the increase in energy costs, and the depleting supply of fossil fuels, the future of constant energy supply is even less certain, and by focusing on consumer behaviour, and changing that behaviour, energy savings could be made without

significant cost or disruption. This method of informing the consumer where they can make energy savings also gives the final say in what changes they make to their daily routines, and they could therefore be more likely to react to the information in a positive way.

The introduction of smart metering can be seen across the world. These smart meters replace the existing meters in the home, which have added functionality and the inclusion of communication devices. By including these communication devices Automatic Meter Reading (AMR) is used as the primary method of billing. These smart meters make up Advanced Metering Infrastructure (AMI), and have allowed the meters to become more than just a metering device, as information about energy usage can now be monitored both locally via consumer feedback devices and remotely by the energy suppliers.

The feedback of information to the user can be customised to the requirements of the user depending upon what their drivers are for reducing their energy usage. This can be customised by either using energy cost as a driver so that it shows that there are cost savings to be made when load shifting at peak times, or by going back to the initial project objectives, there are possibilities that the reduction of carbon emissions can be used. They key issues that surround this are that there needs to be a driver for the consumer to participate in the scheme, and these different drivers can be applied depending upon the situation. Custom profiles can be built using NILM that will allow the monitoring of the energy consumed over periods of time, and therefore an in-depth analysis can be carried out on the data.

This research complemented other work being undertaken, as continual developments within the smart metering area continue. The research was deemed to be an add-on function to smart metering, which would not be required through legislation,

but would be a unique selling point (USP) to smart metering that could be used for breaking into new markets.

The researched concepts were also viable in many different areas, and were not just useable within the domestic sector, which is where the researched has been based, but could also be used within small to medium business, or applied to larger industrial entities. The key identifying characteristics of this approach to NILM is that it negates the need for sub metering to be installed, which can be costly, disruptive and time consuming, and therefore provides an alternative without the need for additional expensive hardware.

1.3. Initial Goals

1.3.1. Load Monitoring, Data Capture and Pre-processing

The creation of a NILM methodology that could be used to monitor the energy usage of loads within the premises required the analysis and capture of loads to build and test models that could be used within the system. A data capture and analysis methodology needed to be created to monitor the loads that were being analysed both individual and the total load current and voltage.

The research required the acquisition of vital information about the current signal of the individual and total load, which would be used for the classification of appliances. Fourier analysis provided an excellent platform for the creation of profiles, as further analysis of the Fourier output provides both magnitude and phase information of the signals. The phase and magnitude information contained from the analysis directly related to the operating conditions of the loads, and could therefore be used to define the appliances though profiles built around the different harmonics.

1.3.2. Load Disaggregation

The main aim of the research was to segregate the different loads being consumed within the premises into their component appliances from the overall load draw into the premises. This is conducted using NILM technologies which have been discussed within the literature review in section 2.2.

By using NILM, the appliances that were being used within the premises were able to be disaggregated using the method known as Canopy Clustering, with its operation discussed within section 4.4. Canopy clustering was able to separate out data into groups efficiently using inexpensive segregation techniques which separate out the data depending upon its Euclidian distance when the fundamental and harmonic content is plotted within a two-dimensional space. By using canopy clustering to separate out the different load groups within the dataset, there was no longer a requirement to know how many different clusters needed to be found, as the process, by nature, separated out the data into the amount of clusters present.

1.3.3. Load Classification

When considering the usage of appliances within a domestic environment for example, it should be noted that there will be multiple loads being used at any point in time, and this knowledge means that an approach to NILM needed to be considered with this in mind. These multiple loads that are present within the dataset are seen as a single group, and therefore the group of loads needs to be disaggregated into its component appliances.

To better understand how these groups of loads were constructed, the interaction between the different load currents should be considered, and used to separate out the individual loads present within the groups. This involved the analysis of the individual loads with respect to the load phase angle and the load magnitude of the current with regard to the applied voltage. Each individual load that is being monitored will have its

own load profile, and by regression the total load can be disaggregated into the individual appliances using the pre-defined load profiles.

1.3.4. Consumer Feedback

The final objective was to be able to provide energy usage feedback to the consumer relating to the appliances used within the premises. By being able to monitor the loads through NILM methods, specific loads may be identified, and the time of use of the energy recorded with relevant energy usage measurements.

1.4. Long Term Goals

By proving a system that was able to identify loads without the use of sub metering allows for a seamless installation of the system into the home without the need to greatly inconvenience the end user. It is envisaged that by building a robust NILM system with the use of canopy clustering, a system could be installed into homes or small commercial environments and be used to monitor and feedback energy usage of appliances and loads within the premises. Combining the information with potential energy savings information i.e. moving loads to different times of the day to prevent excess energy usage at peak times, or to a part of the day with a cheaper tariff, will allow the end users to be able to make informed decisions on how and where energy savings can be made.

The system will be reliant upon known load profiles for it to function correctly and the vision of creating online databases of loads that are present within the domestic environment, could allow for the use of NILM attached to the networks and the internet to access this data. This information could be used to carry out NILM and load identification and profile creation of the appliances within the premises.

Chapter 2. Literature Review

2.1. Introduction

The literature review facilitated a critical analysis of available literature within the chosen research area. This section starts by discussing the core concepts of Non-Intrusive Load Monitoring, and how and where these technologies could be used. The review then follows onto the commercial and domestic uses of Non-Intrusive Load Monitoring and how developments in smart metering could aid in the delivery of the technique.

One of the important concepts within the research was clustering technologies. The literature review itself does not cover the different clustering techniques, as it had significant importance to the research, and therefore has been allocated its own chapter.

2.2. Non-Intrusive Load Monitoring

2.2.1. Introduction to Non-Intrusive Load Monitoring

In the context of this programme of work Non-Intrusive Load Monitoring (NILM) is defined as the monitoring of the energy usage of electrical appliances within the home, or small commercial premises without the need for the direct installation of individual metrology equipment [1], [2]. The use of NILM within such environments allows users to monitor and control how, when and where energy is consumed by using the information obtain through measuring technologies such as current sensors [3]. Traditional methods of electricity metering with electromechanical devices do not provide sufficient information in terms of resources for NILM to be effective, but with the installation of digital metering within residential properties, the functionality available has increased considerably [4]. Electricity meters are no longer simply a means of measuring accumulative electricity usage [5], but now offer greater

operational parameters such as maximum demand within commercial environments, along with the measurement of real and reactive power, and even harmonic distortion within a single meter.

Traditional methods of appliance monitoring would involve the use of sub-meters, which means the inclusion of metering at the electrical switchgear for individual usage zones or even appliances. Sub metering can provide a range of information about the individual appliance, such as power usage and the performance [6], but can also provide other vital information to end users as to where energy savings can be made. From a business perspective sub metering is used extensively for the monitoring of energy usage within individual locations [6], [7].

Sub metering use within the commercial setting has been acceptable for monitoring energy usage within specific departments, but taking this technology into the domestic environment would not be economically viable due to installation costs of the hardware needed and the disruption in retro-fit scenarios. By removing the intermediary step of sub metering and analysing usage data through NILM methods, significant savings can be made with the reduction of sensors and hardware, which is replaced by a meter with built in NILM.

2.2.2. Advantages of Non-Intrusive Load Monitoring

NILM has many benefits, not just for the end user, but can also provide valuable information to the power generating companies as to the habits of domestic users [8], [9], [10] and harmonic content and total harmonic distortion which greatly affects the quality of the power supply [11]. This information could be coupled with demand response programs that offer consumers incentives for reducing their power consumption at certain times of the day, using methods such as flexible pricing [12], [13], [14], [15]. By informing consumers of tariff changes during peak demands, coupled with the provision of information as to where energy in the home is being

consumed, and where energy may be saved if non-essential loads are turned off, load levelling may be achieved [16], [17], [18], [19], [20], [21].

Demand response can be coupled with NILM to aid in the management of loads at peak times, where load shifting of non-essential loads to off peak periods can help reduce the strain and losses on the distribution networks [22], [23]. With the losses related to the square of the current of the resistive loads [24] in the network, it can be seen that there would be significant increases within distribution losses during peak periods. Substantial energy savings could be made by load shedding, with the NILM system being used to aid in the decision making process of determining which loads should be switched off and used at alternative times of the day.

For NILM to be viable in the domestic environment, algorithms need to be developed that are computationally inexpensive due to the hardware restraints and micro-processors that can be used in electricity and smart meters. This will allow NILM to be implemented without the need for separate expensive hardware to carry out the analysis on the consumers' energy profile, and will have a greater chance at being adopted in the industry.

2.2.3. Smart Metering and Domestic Attitudes

Smart metering brings the development of new methods of feeding back information to energy consumers [12], [16]. The proposition is that the implementation of smart metering should be able to provide information to consumers' that was previously not available, and that knowledge of real time cost and energy consumption in the home can help in changing the behaviour of the general energy user [25].

European Union directive (2006/32/EC) [26] has been made to ensure the provision of meters that are capable of displaying correct and accurate information on energy consumption and time of use. These requirements have led to the push forward in the smart metering sector [27]. The planned installation of smart metering means that

a requirement for two way communications between suppliers and consumers will exist and, therefore real time billing may be provided, as well as new methods of billing within the domestic sector [28]. The introduction to smart meters may also provide a new way of communicating with the energy users through the integration of in home displays that communicate with the smart meter directly [29].

Smart meters provide a method of implementing NILM in the home without the need for additional hardware for monitoring and analysing the load profiles of the home. The computation power of the smart meter needs to be taken into consideration when developing NILM algorithms, and need to be efficient enough to run on smart metering devices.

2.2.4. Non-Intrusive Load Monitoring Methods

There are many different types of NILM methods which have been researched previously and have been discussed in the following sections.

2.2.4.1. ON/OFF Power Analysis

The first stages of NILM were to disaggregate the loads into their component appliances. One of the initial ideas about load monitoring involved the monitoring of the turn on and turn off instance as measured within the power draw of the property. By monitoring the appliance turn on and turn off power and the duration that the appliance was in use. Energy information could then be obtained and recorded against the appliance and used for monitoring purposes. The use of monitoring the on and off switches of the appliance was where NILM began [30].

NILM has progressed significantly since its initial inception by Hart [30], where it was devised as a method of removing sub metering to monitor a collection of appliances. Initial methods relied on the monitoring of on/off instances within the power consumption period [31]. Many typical appliances within the home could be categorised

as on/off devices, such as lighting, and analysing the power signals of such loads within the time domain, will show appropriate increases and decreases of energy consumption as being indicative of the load operational periods [32], [33].

By comparing the increase and decrease in power consumption, the time that the load is in use can be calculated, and the total energy consumed is then determined. Load profiles of classes of devices can be created around the assumption that equal increases and decreases of power consumption could be attributed to the same load, though identification of fundamental loads within a group is not trivial. This statement holds true when the examples of resistive loads are monitored such as a kettle. The kettle only has one operating condition, which can be said that the kettle is either on or off, but the duration that the load consumes energy will vary depending upon the wattage of the element for any given mass of water.

When NILM was originally postulated, appliances within the home were relatively simple in their operation, and most of the appliances that needed to be monitored would have been resistive. Hence the monitoring of the on/off instances of the appliances was considered appropriate at that point in time [30]. The application of this method when applied to today's domestic environments, give rise to many issues that need to be tackled. With the onset of the digital age many homes now have a vast range of electrical appliances and consumer electronics such as computers, DVD players, mobile phones and various other entertainment and domestic "white goods". Most of these new loads are non-linear [34], [35], and their energy consumption will change over time, making the matching of on/off instances more difficult with the amount of loads present within the home.

The method described by Hart [30] was only viable for use with larger household appliances, due to the nature of the power consumption, and detecting smaller step changes in energy consumption while applying noise filters can become

difficult. Cut off levels need to be determined within any algorithm for the decision for loads being switched on or off, and Hart's paper describes a 15W tolerance which would be too large when taking many of the consumer electronics into account that are being used today. This high level of tolerance used for the cut-off thresholds may be useful for the utility companies, as demand response programs are only going to take into consideration the larger energy consuming appliances such as vacuum cleaners, washing machines and ovens for example.

Taking the above into consideration, Harts methods may no longer be viable in today's environment. The amount of non-linear loads that are being used has increased and the currents in these loads become small in comparison to the larger white goods within the premises. Therefore the cut off tolerances that have been described are no longer adequate for the application of NILM, and may only be viable for monitoring larger scale appliances, with simple operating characteristics as smaller power consuming electronics draw less current than the cut-off threshold [34].

Furthermore, there are instances where using the on/off characteristics for real, reactive and harmonic power draw are inappropriate, which points to the use of transient detection algorithms being employed [36]. This method of load classification takes the approach of using pattern recognition to match start up characteristics of the loads being used. This was an important step forward, as more complex loads were able to be classified from the component make up, such as charging capacitors or heating elements within the appliances being monitored.

One of the advantages of using on/off power matching is that the computational overhead is small, and the algorithm only involves the computation of the increase or decrease in power levels. Once these have been calculated, the value of the power change is then matched to a pre-existing list of appliances in the property. The low

computation of the process makes this form of NILM ideal for implementing into domestic smart metering devices.

By using on/off power analysis of loads only, a lot of information would have been omitted from the analysis such as the harmonic content of the appliance and how different appliances interact with one another. This approach narrows the scope for more detailed analysis of the problem and therefore the system needed to be expanded to allow for accurate NILM within domestic and commercial uses of the present day and a more fine grained approach would be required.

The use of EMF sensors has been used in the area of NILM for the detection of the state of appliances [37]. Even though the use of sensors is traditionally not used in NILM, EMF sensors have been used for the monitoring of the state of the appliances to aid in the training of the NILM system. With the EMF sensors being placed around the current carrying conductor, the system can still be classed as a non-intrusive method of load monitoring.

The system uses a combination of Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA) clustering techniques. By applying PCA to the results obtained from the EMF sensors, the features with the highest variance were able to be extracted and used for classification, whilst removing the features that provide correlation between the features [37]. By applying LDA to the results from the PCA, the results are displayed on data axes that highlight the interclass variance whilst keeping the intra-class features [37]. Further analysis on the dataset is conducted in the form of k-means clustering which defines the final groups.

Due to the use of EMF sensors, the system will be subject to external noise from nearby appliances, and will therefore only be useful for the monitoring of the states of the appliances that are in use.

One of the drawbacks of the this approach is that there is no form of current and voltage measurement, and therefore vital classification information such as phase angle will be lost. The type of load may be found and classified into two categories such as linear and non-linear loads which add another dimension to the classification profiles.

Automatic classification still requires prior knowledge of the loads that are present in the system. Further work is being conducted to implement the use of neural-networks and support vector machines.

2.2.4.2. Rules Based Disaggregation

Rules based disaggregation provided a method of monitoring the behaviour of the energy user. The method of disaggregation adds another dimension to the analysis and classification of appliances in NILM, but also has the advantage of being able to monitor user behaviour. User behaviour can aid with customer feedback which is a key feature of a NILM feature. By monitoring energy usage patterns in the home feedback can be given to help change behaviour and make energy savings.

Behavioural patterns of the end user provide a significant and important consideration within the NILM process. Individual domestic homes will have an energy profile, where certain household tasks such as cooking are carried out at specific times each day [38]. This allows a usage profile to be constructed for the household, and by coupling this with other information such as consumer appliance ownership, which will include operating conditions of the loads, more sophisticated monitoring methods could be envisaged. Using techniques such as decision trees [38], the loads being consumed within the household can be logically determined. Such an algorithm would take into consideration the usage of loads within the premises on the assumption that, for instance, air conditioning may be on at certain periods of the day, and that, coupled with the increase of energy consumption equivalent to the power draw of an air conditioning system, would be enough for the algorithm to determine the correct load allocation. The

method would be simple enough to include in current smart meters due to the low overheads of the algorithm.

By using a rules based algorithm, larger loads could be disaggregated with an average of 7% difference to the peak load consumption [38]. This could be achieved by monitoring the duty cycle of the appliances that are being used, and the algorithm would be developed for simply disaggregating large scale appliances within the premises such as water heaters and air conditioning units, and unable to identify any smaller loads being used. This approach would allow for end user feedback by monitoring energy usage patterns but falls short when it comes to monitoring a wide range of lower energy consuming appliances or devices in the domestic environment therefore limiting its use within NILM to larger scale applications.

2.2.4.3. Neural Networks.

The previous methods of NILM have been focused on the larger appliances and have been limited to using the power consumption of the appliance for disaggregation. To be able to further the analysis of the loads used within premises further information needed to be obtained such as the harmonic information which allows for a more in depth approach to analysis, and also allows for smaller loads to be classified using the extra information obtained through harmonic analysis.

Harmonic analysis could provide new ways for load disaggregation to be realised within the field of NILM [39], [40]. By considering the harmonic signatures of individual loads [41], [42], key identification aspects may be extracted which could be used for classification of the loads. The harmonic information would need to be analysed, and methods such as Neural-Networks provide a great opportunity for harmonic classification [43]. The neural-networks described are able to distinguish the loads present within the premises by analysing both the real and imaginary harmonic components contained within the current draw.

Neural-Networks were developed around the representation of the relationships of functions with inputs and outputs in a similar way to how neurons in the brain are used in Biology. Within the uses of NILM, the inputs to the system are represented by the harmonic data of the loads that require classification, and the outputs of the system are the loads being classified. The inputs and outputs of the Neural Network would be connected by nodes that represent the functional relationship of the input variables and output loads, and could be programmed with the use of learning algorithms and training sets of data. The training process sets up the hidden nodes within the system, assigning weights to each of the nodes. An example of the system can be seen in Figure 2-1[44].

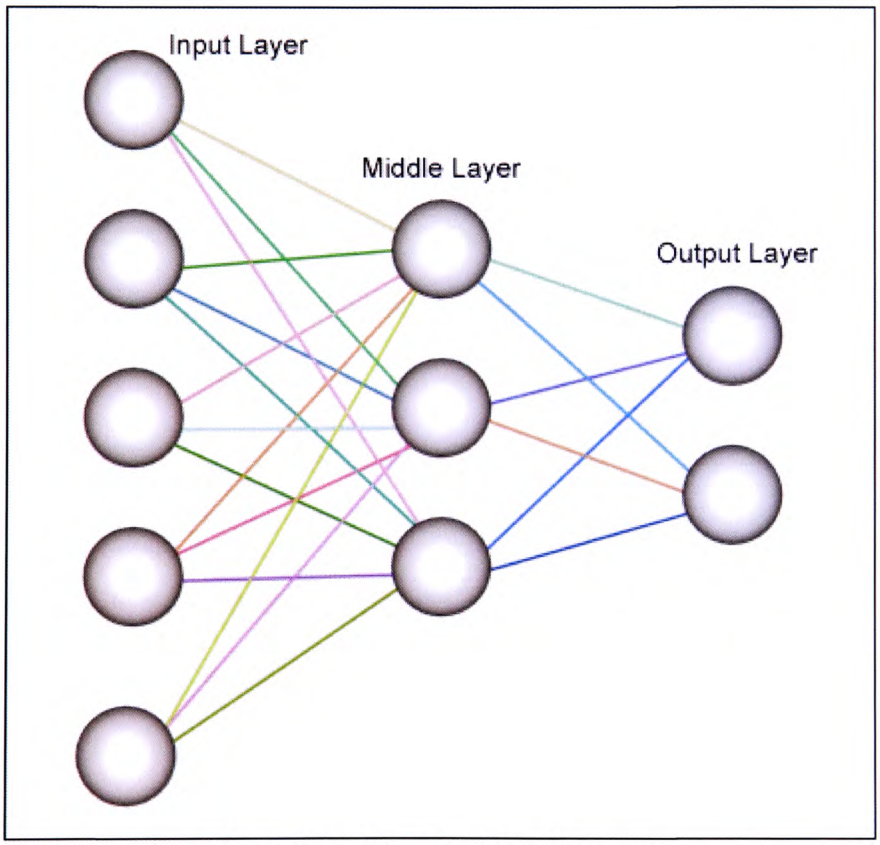

Figure 2-1– Neural Network Layout

These are used for pattern recognition and the use of the Neural-Networks within NILM provides a good basis for analysing the harmonic signatures of loads [45], [46], [47]. Within this example the inputs to the system are the harmonic values; both

real and imaginary components are taken into consideration and power factor may also be a consideration [48].

The artificial Neural-Networks are adapted from the biological neurons that are present within the human brain. The inputs and outputs of the networks could be determined depending upon the amount of harmonics within the system that are being used for classification, and the number of outputs will depend on the number of devices being classified.

By using Harmonic analysis, the Neural-Network approach was able to correctly identify loads that it had been trained to identify through representation of the harmonic content contained within the current signals [49]. The advantage of using harmonics to create a profile could be seen when monitoring different models of a particular device, such as various models of LCD monitors. These devices even though perform the same function, are represented differently within the frequency domain due to the different component make-up of the various models. Conversely multiple devices of the same model can be seen to be the same within the frequency domain, allowing for noise, therefore making profile creation simpler as the same profile can be used for multiple devices of the same model.

The analysis of harmonic signatures has shown that there needs to be some consideration when creating profiles for the noise within the system [43]. Models that are created need to consider the noise in each of the harmonics that may be present, due to operating conditions and even the measuring devices. Building profiles via the application of these noise assumptions can greatly improve the success rate of identifying the different loads within a NILM system.

As with most methods of NILM, the neural-network approach also requires training to be completed [50], [51], [52], [53]. Training can be carried out by using either laboratory results, or using mathematically deduced training sets. Laboratory

training will be conducted using the profiles of real loads that are present within the system, which would be representative of the loads being classified, whereas the mathematically deduced profiles are created around the assumptions of the profiles that may be seen within the system. One issue with this is that as there are a range of different loads within the home, there could be more than 10 different loads to be classified [54]. This is where training of systems can become an issue, by looking at each of the loads in a binary state of either on (1) or off (0), the different combinations of loads can be calculated as 2^{10} combinations. Due to the amount of combinations of loads that need to be identified, there is a large increase in the work that is required to complete load disaggregation at a high computational cost, which therefore restricts its uses within domestic NILM.

To alleviate the issues surrounding training of systems, mathematically deduced training sets may be deduced for individual appliances. This method has proved useful within NILM systems, and in particular within neural-network systems [43]. Although mathematical datasets aid the training process of NILM it will not be able to account for the number of products and appliance that are in use which need to be disaggregated, and therefore a custom approach to training depending upon the premises that it is installed would be required. Due to the extra work needed to initially set up a NILM method based on neural networks, current smart meters would not be capable due to the small processing power compared to desk top units which would be needed to initial train the system, and makes the use of neural networks more difficult to implement in the home.

The use of Neural Networks within NILM has shown that loads could be disaggregated using other information obtained from further pre-processing of the loads energy consumption by converting the data into the frequency domain. This has opened

up many more characteristics of the appliance that can further be used for the classification of the loads within the frequency spectrum.

2.2.4.4. Commercial Uses of NILM

The research has been based on NILM in the domestic sector, but previous studies have shown benefits of NILM in the commercial sector, and these ideas could be looked at and implemented within domestic premises.

From a commercial perspective the same principles such as steady state analysis and transient analysis can be applied. There are many commercial requirements for monitoring appliances and loads within a business such as building management systems. When monitoring these systems there are large potential costs that could be incurred, from sensor costs to installation costs. It is clear that the use of power signature analysis in the commercial environment would greatly aid with process of NILM [55], [56], [57], [58].

2.2.4.4.1. Steady State Analysis

The use of analysing the power signature of the turn on/off instances can be used to determine which loads are present [59]. Within an industrial environment there are many systems that display real and reactive power consumption profiles and these instances can be plotted in a two-dimensional plane as seen in Figure 2-2 [55]. By matching the equivalent step changes, both increase for turn on and decrease for turn off in these planes, the durations of the loads can be determined, and therefore the total energy consumption.

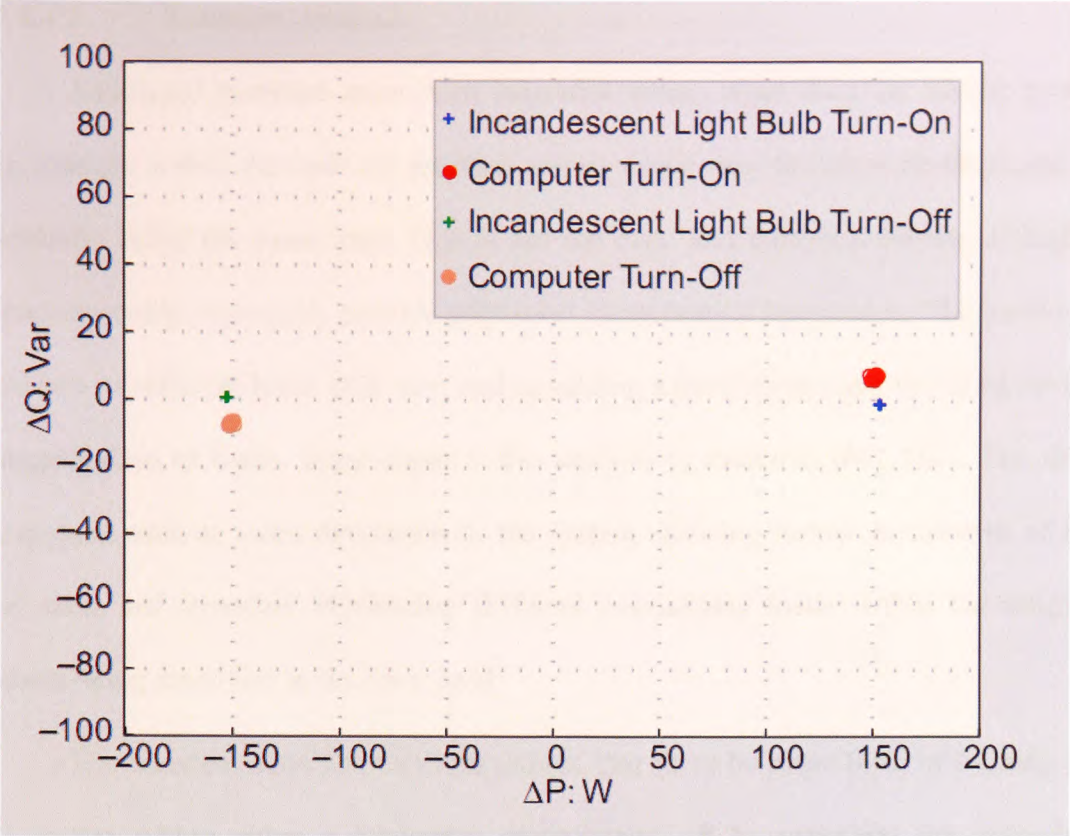

Figure 2-2 – Power Signature Analysis, Real and Reactive Domains

The steady state analysis of the power spectrum shows that classification of loads can be achieved using multiple variables of the signal such as the real and reactive power in a two-dimensional domain. This method can be applied using other variables such as harmonic content of the signal, where the points of interest will be the increase and decrease of multiple harmonics from the appliances turning on and off.

This method follows the initial principles described by Hart [30] but uses the different power domains to provide additional feature sets for disaggregation, and will have similar computational requirements of Harts methods. The low computation requirements of the algorithm are ideal for the implementation of the algorithm within smart and home energy meters making for the system to be easily installed and implemented with minimal end user disruption.

2.2.4.4.2. *Transient Analysis*

Additional problems arise in an industrial setting when there are similar power step changes within the real and reactive power, which may therefore be displayed as essentially being the same load. This is not the case, and therefore the use of higher harmonics could potentially provide additional identification knowledge. The harmonic signature of different loads will vary, and by adding a third component to NILM for the disaggregation of loads, more depth to the analysis is available [60], [61]. The third component adds an extra dimension to the system allowing further breakdown of the load data, and therefore eliminating different overlapping loads within the original domain being classified as the same load.

The transient detection methods require that there be some form of training for the system within either a laboratory environment, or by recording the individual transient components of specific loads to provide an exemplar for use in the identification process. Particular examples of usage within industrial sectors include large scale Heating, Ventilation and Air Conditioning (HVAC) or significant sized motors. This form of transient analysis evaluates transient harmonic distortion patterns generated by a system that is repeated at the start of a process such as the ramp up of the motor [36].

This method of NILM provides real time analysis of the appliances within the premises, by continually comparing windowed samples of power consumption within the time domain to the exemplars, and using a least squares methods to calculate if there is a match. The sensitivity of the categorisation can be determined by setting the threshold limits of the least squares analysis, and should it be within this threshold the load will be classified [36], [62], [63].

Due to the heavy use of induction motors and HVAC systems, commercial building management and industry are prime candidates for the use of transient

detection within NILM. The turn on transients of motors can vary across many applications, but may share similar start up patterns. Due to this transient analysis can be used for different load dependant, time varying transients of the same pattern [36]. This is also applied to varying power loads that share the same transient profile. By scaling the exemplars through time and power domains, similar loads may be uniquely detected and classified by their transient start up patterns [64] [65]. An example of an induction motor in the power domain can be seen in Figure 2-3 [62].

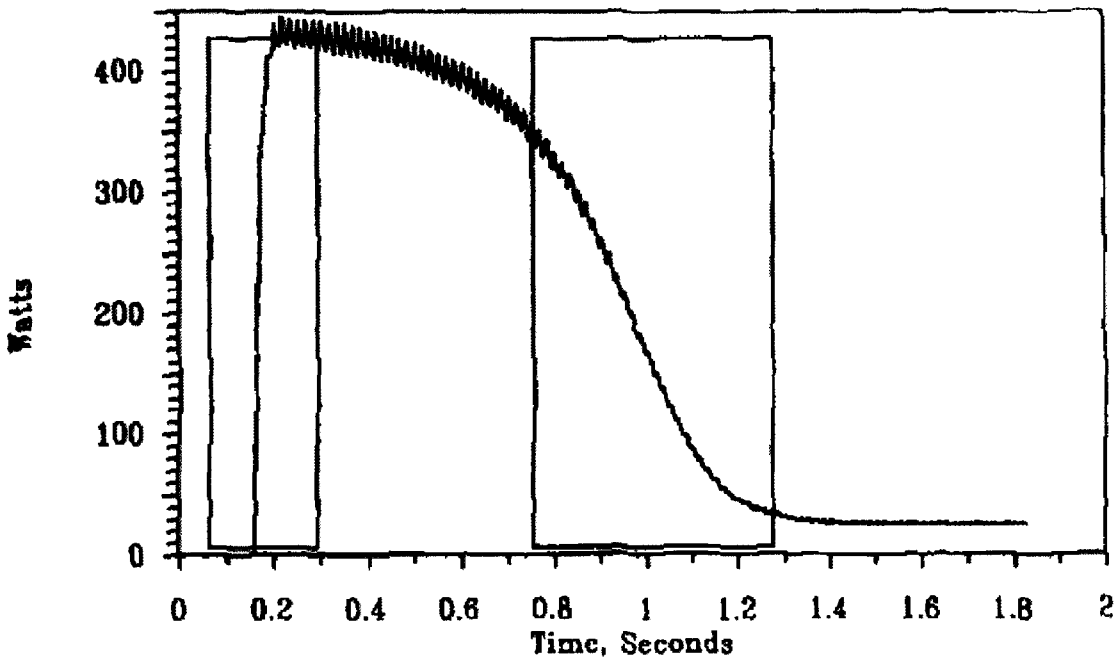

Figure 2-3 – Real Power Transient Start-up of Induction Motor

One of the downfalls of the transient analysis approach could be seen when there was no turn off transient where loads just suddenly stop. When a load ends in this manner, it was seen as a drop in power, and there were no patterns to be seen for matching purposes. This posed a problem when there are many loads of the same power rating being monitored, and for this transient analysis requires the integration of other NILM methods to be able to identify these turn off instances.

Further investigation into the use of transient analysis has shown that there is a potential for a hybrid approach to be taken which involves using both steady-state and transient analysis in NILM [62], [66]. Transient analysis alone can be very computationally intensive in nature, and by combining two methods of analysis this can be reduced. Leeb et al [62] proposed that by using a sliding mean, transient analysis would only be conducted on sections of data where there was a significant increase in the mean power consumption, which indicates the turn on of a load within the system. Taking turn off instances into consideration, there was a stoppage of power consumption and therefore the transient becomes unusable since there is no transient to the turn off of the load, and thus steady state changes can be used to discriminate which loads are turned off.

When trying to apply transient analysis to the domestic environment, additional complications would be found in that there are many appliances in the home that are simple turn on/turn off in their operation. By adding transient analysis the problem becomes overcomplicated for the home as there would be many different transient patterns that would have to be learnt, and the process would have to be coupled with other forms of NILM such as steady-state. Transient analysis implementation within the home also becomes difficult due to the amount of resources needed to constantly monitor and match patterns obtained through monitoring. The large amount of processing that is required means that the algorithm is more suited to being run on a PC rather than in a smart metering device and prevents this method from being used in the domestic environment.

2.3. Conclusions

The Literature review has shown that various versions of NILM strategies have been used to monitor energy consumption within both domestic and commercial/industrial environments, with varying degrees of effectiveness. By using NILM strategies the need

for sub metering would be removed, and therefore the impact on the end user when retrospective installation needs to be undertaken is minimized.

There are many different techniques that can be used within NILM. The techniques have progressed from the simplest ideas such as monitoring step changes within the power consumption, to more advanced techniques such as harmonic analysis.

The development of NILM from its inception has mainly been concerned with monitoring of large appliances and white goods, and the use of monitoring on/off instances was sufficient to build a NILM system during this period. Due to the ability to monitor only larger appliances and white goods, this current system is no longer viable when finer grain load monitoring is required.

With a change in consumer attitude towards the reduction of energy consumption within both domestic and commercial sectors, using traditional methods of NILM as suggested by Hart [30] reduces the amount of devices that were able to be monitored, and as such there was a requirement for more in depth analysis to be conducted within NILM for greater refinement to be achieved.

Using methods such as pattern recognition has enabled the use of transient analysis to be incorporated into NILM [67]. The methods of analysing the start-up transients of loads has a limited usage within the domestic sector, but when considering the industrial sector there are advantage that transient analysis bring to NILM that cannot be ignored. Many industrial sites comprise numerous similar machines that can be monitored through NILM. These range from frequency inverters to induction motors. When considering monitoring induction motors through NILM, there was one key identifier that can be used to build a profile, and that is the start-up transient. It should also be noted that the start-up transient of a motor is common across many different size motors, and therefore a single profile can be used for identifying the motor start up transients. The only differentiating factors that need to be considered are the start-up

time and the maximum operating currents, and these can be scaled from the expected motor start up transient profile.

The transient analysis method alone is only useful for when loads have a turn off transient, but many loads will just stop operating and consume no current when the load has been turned off. The literature review has shown that transient analysis alone is a poor choice for NILM, and will need to be coupled with other NILM techniques to prove useful.

By using harmonic Analysis, there is the scope for loads to be classified using a large number of variables such as the individual harmonic components to create a profile of the loads. Each load will produce an identifiable current signal within the time domain. Converting this signal into the frequency domain, the harmonic content of the current can be found and used to create load profiles for load identification within NILM. These harmonic signatures can be used for the classification of loads using the various different methods of NILM such as Neural Networks or steady state analysis.

The literature review has also shown that monitoring the harmonic signatures of the current signals produced by the loads provides the most information of the load that can be used for classification purposes. There are ways in which this data can be analysed either through domain specific analysis as proposed by Laughman et al [55] or Neural Networks [43], which both provide suitable results for disaggregating a number of loads, but fail to identify loads that have different operating conditions.

Other techniques have been found that aid in the classification of loads within NILM such as rule based algorithms [38], which are able to identify loads based on their operating characteristics. While this method proves sound with larger energy consuming devices, it still does not cover all of the loads that need to be monitored within the scope of NILM.

Careful consideration needs to be made when deciding which algorithms to use in NILM, since there are limitations in the hardware that is used for NILM. NILM systems will be developed for use within smart meters, and therefore the algorithms need to be able to run on the smart metering hardware without affecting the metrology of the device.

It is apparent from the literature review, that research into NILM has progressed from its initial inception, and more robust methods of load analysis and classification have been continually sought. Each of the methods described provide clear defining arguments as to why they should be used within specific areas of NILM due to the nature of the problems being solved, and the approaches used. This still leaves the area without a complete solution that covers a range of loads that need to be classified, from the large energy consuming appliances for example washing machines, down to smaller non-linear loads such as televisions.

The key concepts that have come from the literature review are the use of harmonic analysis, which should greatly improve an in depth analysis in to what is happening within the premises from the electricity consumption perspective.

Chapter 3. Technology Review

3.1. Overview

This chapter aims to introduce the specific technical aspects of the research that are significant to the experimentation. It considers the various current sensing technologies that may be used within a NILM system and how they may be integrated. These include current transformers, Rogowski coils and Hall Effect sensors. A further review has been conducted on the hardware used within the research for the data collection, and the statistical software applicable for analysis.

3.2. Current Sensors

With the use of NILM in the domestic environment mainly reliant on current sensors being installed into the metrology system, a range of current sensing technologies have been reviewed. The three main technologies that are discussed as being deemed usable within NILM and within the domestic environment are:

- Current Transformers
- Rogowski Coils
- Hall Effect Sensors

3.2.1. Current Transformers

Current transformers have historically been used extensively within the electrical and electronic industry for measuring currents within appliances and for energy metering devices. One of the main advantages of this type of sensor is its galvanic isolation, and many different types are found, such as clip on transformers which can be used when the circuit cannot be broken, therefore a sensor may be placed around the current carrying conductor.

The current transformer measures the current within cable through its magnetic induction, B, to a secondary coil wound round a magnetic core. The ratio of the current induced within the secondary coil is proportional to the ratio of the number of windings on the secondary coil to the primary coil which can be seen in Figure 3-1 [68], and the relationship described in (3-1). The final reading from the sensor is the voltage across a shunt that connects the two ends of the coil.

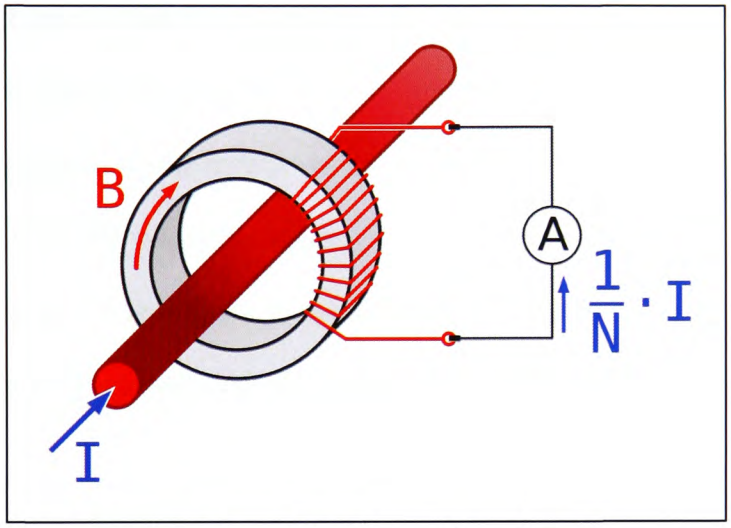

Figure 3-1 – Current Transformer

$$I_{out} = \frac{1}{N} \times I_{in} \quad \text{A} \tag{3-1}$$

Current transformers are relatively low cost devices, and they can be found in a range of accuracy classes, with a typical accuracy range of 1% to 10%. Due to the current transformer being based on an iron core, saturation problems may arise. This is an important consideration when deciding on the current transformer ratings relative to the currents being measured. The current transformer should be chosen to cover the full range of currents that are to be measured by the system; otherwise currents that are greater than the current ratings will put the transformer into saturation, greatly reducing the accuracy of the device.

3.2.2. Rogowski Coils

The Rogowski coil, like the current transformer, is a sensor that is placed around the current carrying conductor, and is also a coil based transformer sensor. The main differences between the two are that the Rogowski coil is based on an air based coil [69], [70], . Again with the Rogowski coil the current induces a current into the coil through magnetic induction through the air core, and the amount of current induced is smaller than that seen within the current transformer due to the lower permeability of air (μ_0) compared to that of an iron core.

The relationship of the output voltage to the input current can be determined by equation (3-2) [71]. The voltage is equivalent to the rate of change of the current, and therefore to obtain a viable voltage output that is representative of the voltage and integrator circuit [72] is attached to the output of the coil as seen in Figure 3-2 [73].

$$\int_0^l v dx = -\mu_0 N A \frac{\delta J}{\delta t} \text{ V} \qquad (3\text{-}2)$$

The main attributes that determine the Rogowski coil are the length, l, the cross-sectional area, A, and the number of turns, N, making the Rogowski coil highly customisable depending upon the area of application. Rogowski coils have found many uses in the areas of high voltage and current [74], [75], [76], [77] and high frequency applications [78], [71], [79], [80], [81].

One of the main issues surrounding the Rogowski coil due to the air core is external interference, which greatly reduces the accuracy of the device. The accuracy of the Rogowski coil is also affected by the positioning of the coil with respect to the current carrying conductor [69], and therefore how the coil is installed needs to be considered, and modifications to the design can be implemented to position the conductor in the centre of the coil. This however, restricts the way in which the coil can

be integrated into existing circuits and there has been some research that suggests using electrostatic shields to prevent external electronic interference [82]. Comparing the Rogowski coil to that of an iron cored current transformer, the accuracy of the Rogowski coil is greatly reduced at the lower currents [83], which are used within NILM due to the effects of noise on the circuit, and lends itself to be better suited for high powered electrical distribution networks [84]. By making the transducer more sensitive within the operating limits of NILM, the influence of noise may be reduced, but as a result the transducer will become bigger in size, making the device more intrusive when used for NILM purposes.

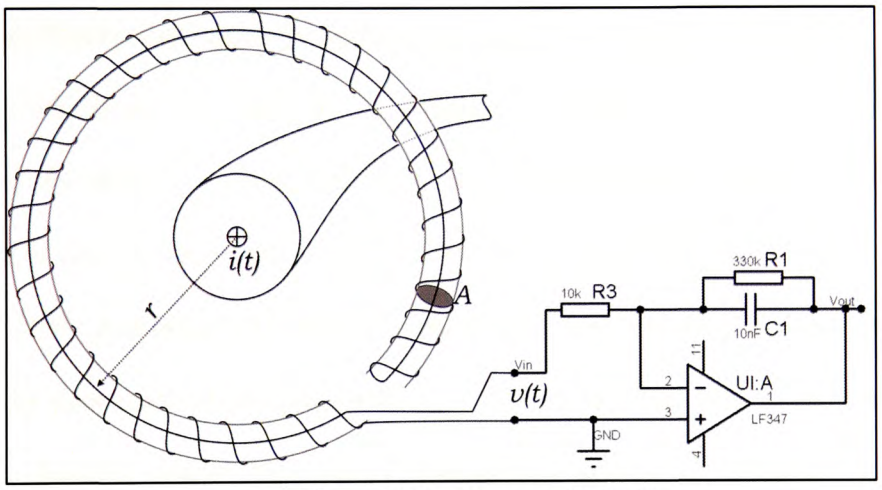

Figure 3-2 – Rogowski Coil

3.2.3. Hall Effect Sensors

Hall Effect sensor takes a different approach to measuring the current as the sensor does not surround the conductor, but is instead placed next to it. This form of sensor is an integrated circuit (IC) based device and uses the Lorentz force as an indicator to the amount of current being measured [85]. The sensors need to be powered as does the Rogowski coil amplifier meaning an external power supply is required.

The Hall Effect sensor measures the current through the conductor by the forces of the magnetic field produced by the current. The hall sensor itself has a current

passing through it, and the Lorentz force stipulates that the change in magnetic field perpendicular to the current passing through the sensor will alter how the electrons move through the sensor, therefore changing the Hall voltage [86]. The Hall voltage can be determined from equation (3-3) [86], where V is the Hall Voltage, R_H is the Hall coefficient, I is the Current passing through the sensor, B is the flux density with t being the thickness of the sensor.

$$V = \frac{R_H I B}{t} \text{ V/A} \qquad (3\text{-}3)$$

3.2.4. Current Sensors Considerations

All three current sensing technologies have been considered within the application of NILM. Specifically looking at the operating conditions of each of the technologies there are clear distinctions that need to be considered when choosing the sensor. When considering the usage of current sensors in the domestic setting, the Rogowski coil will not be a valid choice, due to the noise that is present at low frequencies and low currents which are 50Hz and below 50 amperes respectively and the positional requirements of the coil. The properties of the Rogowski coil lend themselves well to the requirements of current sensors within the power distribution sector, or industry where high frequency currents are being monitored. The current transformers and Hall Effect sensors in contrast provide better sensitivity to these ranges due to the presence of an iron core within the current transformer, and the ability to mount the Hall Effect sensor in series with the current carrying conductor.

For retro fitting purposes both the Rogowski coil and the current transformer are available in clip on variants which enable ease of fitting into existing systems but positional requirements of the coils need to be considered. The Hall Effect sensors are not so easily fitted within a retro-fit environment due to the fact that they require

installation in series of the conductor, which would have to be completed by a trained professional.

The cost of the sensors are comparable for the function that they serve, but one of the main advantages of each of the sensors is that they can be built depending upon the operational requirements of the sensor and the application in which it is being used. With regards to the Rogowski coil, to enable the sensitivity needed for the operating conditions of domestic NILM, there is a requirement for a larger coil, which induces large costs. Both the Hall Effect sensor and the current transformer have reduced costs for the lower measurement ranges and a good cost to accuracy when compared to that of the Rogowski coil.

3.3. Data Acquisition & API

A data acquisition specification was drafted from the ideal design idea that flowed from the literature and sensor survey. A development and test station needed to be defined. A market survey showed numerous possibilities, and the following devices has shown good price/performance characteristics.

3.3.1. PCI-Base II & LIBAD API

PCI-Base II [87] is a hardware data acquisition device with the ability to add plug in modules depending upon the application created by the manufacturer 'bmcm'. Plug in modules, MAD16F [88], are available for the PCI-Base II which provide up to 32 analogue channels for data acquisition with a 16bit resolution and variable voltage range from ±1V to ± 10V. The device is able to read digital and analogue values, and provides a standalone software application for viewing signals. The operating characteristics of the MAD16F can be seen in Table 3-1, and were considered to be in excess of foreseeable requirements.

Sampling Rate	500kHz
Minimum Sampling Period	2μs
Measuring Ranges	±10V, ±5V, ±2V, ±1V
Typical Noise within relevant measuring ranges	±5LSB, ±7LSB, ±8LSB, ±8LSB
Relative accuracy in the measuring ranges	0.0015%
Resolution in the relevant measuring ranges	16 bit (=0.3125mV at ±10V MR)
Channels	16 single-ended

Table 3-1 – MAD16F Operating Conditions

The manufactures have provided a programming API that allows the collection of data via a C/C++ application.

3.3.2. USBAD & LIBAD API

As with the PCI-Base II, the USB AD [89], another data acquisition device, has been developed and manufactured by 'bmcm', but now only requires the use of a USB port, meaning it can be used on a laptop or PC. The functionality of the API provides the same access to functions as the PCI-Base II, but the USBAD is limited in the number of channel to 16, and has a 12bit resolution, and a fixed voltage rage of ±5V.

3.3.3. Software Development Tools

The foreseeable software development requirements indicated that C/C++ would be a significant requirement. The University could provide Microsoft products as standard and Microsoft Visual Studio was evaluated alongside Eclipse, which is an open source equivalent.

Microsoft Visual Studio provides a robust programming environment with access to debugging and other functions such as subversion control through the addition of add-ons. This propriety software is available for free using the express edition, or the professional version via the academic alliance programme which the author is a member

of. The software provides an easy to use interface, and the program set up is straight forward with the use of the project wizards.

Eclipse provided a simple to use, open source IDE which incorporated add-on system to further expand its functionality. Project creation has completed through the in application wizards and was a step by step process. With native support for many languages with pre-built packages dedicated for C/C++ or Java, the software also provides support for many different operating systems such as Linux and Mac OS X.

3.4. Statistical Analysis Tools

The program plan that developed from the initial investigation highlighted the need for a variety of statistical analysis tools. MATLAB and associated toolboxes were considered due to both availability and a previous competence.

3.4.1. MATLAB

MATLAB [90] provided both an IDE and high level language usable for the analysis and exploration of data. It had been well known and widely used in many different research fields and allowed the installation of many different toolboxes depending upon the specified research area. The software provided many methods within the default package for displaying, exploring and graphing datasets and results as well as many functions for data manipulation.

Many different data formats were supported such as data files and Excel spread sheet, meaning data import and manipulation was seamless. Application development was also supported, allowing specific applications to be made with custom built user interfaces, enabling tailored data analysis tools to be easily built to aid with data analysis.

3.4.2. Statistics Toolbox

The standalone version of MATLAB included many graphing and analysis tools, but there were functions that were only available within the toolboxes. The statistics toolbox [91] provides access to other functions and graphing capabilities such as dendrogram plots, grouped scatter plots and the ability to visualise and explore more robust data fitting models.

3.5. Conclusions

The technology review highlighted areas of interest that needed to be considered when conducting the research. Considering the different current sensors there were few issues that need to be evaluated, such as how a NILM system would be implemented. NILM systems that were to be retrospectively installed within the home would need to be done so with minimal intrusion to the consumer. Taking this into consideration, two of the sensors discussed could be installed without the need to break the electrical circuit these being the current transformer and the Rogowski coil, they come in clip-on form. The Hall Effect sensor could also be installed by being placed next to the wire, but when considering the sensitivity of the Hall Effect sensor needs to be placed in close proximity to the conductor to achieve good results. To combat this, packages have been developed that require the breaking of the circuit so that the conductor runs through an IC package and very close to the Hall sensor [92], meaning that Hall Effect sensors are not an ideal candidate for use within retro-fitted NILM.

The technology review has also shown the need for certain hardware and software to be used for the completion of the research. The research is based on real life uses of NILM, and as such empirical data would be used for the building of load profiles. A consideration such as the amount of channels, accuracy and resolution of the channels was an important factor in the hardware decision. The API's provided with the

hardware are robust enough to complete the data logging and have the ability to be able to custom set up the hardware at time of testing.

Determining the IDE to develop an application to capture the data from the DAQ depended on how the API's had been written and their support. There were two main IDE's that had been investigated which were Eclipse, and Microsoft Visual Studio. Eclipse was an open source IDE that allows the installing of third party add-ons. Conversely Microsoft Visual Studio 2010 was proprietary software but also allowed third party add-ons. Both IDE's provide the required functionality to complete the task, and with the use of Microsoft's Express edition there are no cost implications for using either IDE. The decision was made to use Microsoft Visual Studio, due to availability and prior knowledge of the application.

The decision to use MATLAB for the majority of the research was based on the authors' previous experience with the software, and the online support for the system. MATLAB offered all the available tools that were required for the completion of the research and with addition of expanding the package with toolboxes was also a major advantage of the software. One of the main features of the software was the graphical user interface function that provides tools useable for the development of custom user interfaces to enable analysis and experimentation of the data as appropriate to the research.

The statistics toolbox was an addition to the package that was added due to the missing statistics functions included with the standalone software which were required for the research, such as a robust curve fitting tool. There were also many other functions that could be useful for the research that were used with regards to the visualization of the data, and the statistics toolbox offered improved functionality within this area.

Chapter 4. Clustering Review

4.1. Introduction

This chapter considered the different clustering methods that have been used within the research. Clustering gathers data into groups of similarity and is an important part of the research. With data being represented in multi-dimension space, the use of clustering becomes important within pattern recognition and profiling of data.

4.2. Clustering Implementation within NILM

Due to the nature of the NILM which was required to be conducted at the premises where the monitoring is being taken place, the monitoring was likely to be completed within the metering devices itself. This posed problems when the clustering of the data was completed due to the low power of the processors included in meters. The clustering process was only completed at the time of training the system, which would be completed at the time of installation and when new devices were being added to the system. Due to the amount of computations that needed to be conducted during the clustering process, a number of solutions could be proposed such as clustering the data at remote server farms or on local computers within the premises, which would reduce the strain on the metering device.

There were some solutions that could be considered when processing the data for clustering where by the processing units on the meters themselves could be used, or the data could be processed externally. Processing the data locally would put extra burden on the metering device, but due to the clustering process only occurring during the training periods, remained a viable option. Processing the data externally would enable the training of the system to be completed at a server farm for example, where technologies such as parallel computing could be utilised, which could speed up the

training periods. Should the data be processed externally there would be a requirement for some form of internet connection to be included, but with the moving forward of smart metering [27], [93], this was expected to be included with future developments.

4.3. K-means Clustering

K-means clustering is the method of grouping data together based on a Euclidean distance metric [94]. For the k-means algorithm to be used, the number of clusters, k, needs to be known, and the process is based around minimising the distance of all the points assigned to the cluster to its centre until convergence occurs [95].

The algorithm implemented an iterative process and involved an initial stage of measuring the distance between all the points and the cluster centres. The data points were then assigned to the nearest cluster centre until each of the data points was assigned. The average of all the clusters was further determined and the new cluster centre calculated. The process was then repeated until the cluster centres converge, the flow chart for the algorithm can be seen in Figure 4-1 [96].

K-means clustering was one of the simplest clustering methodologies that were available, but as a result of its brute force approach it was very computationally intensive, which for large datasets begins to become an issue. The initial cluster centres needed to be selected carefully during the initial stages of the algorithm, as poorly selected centres could yield a sub-optimal fit for the data [97].

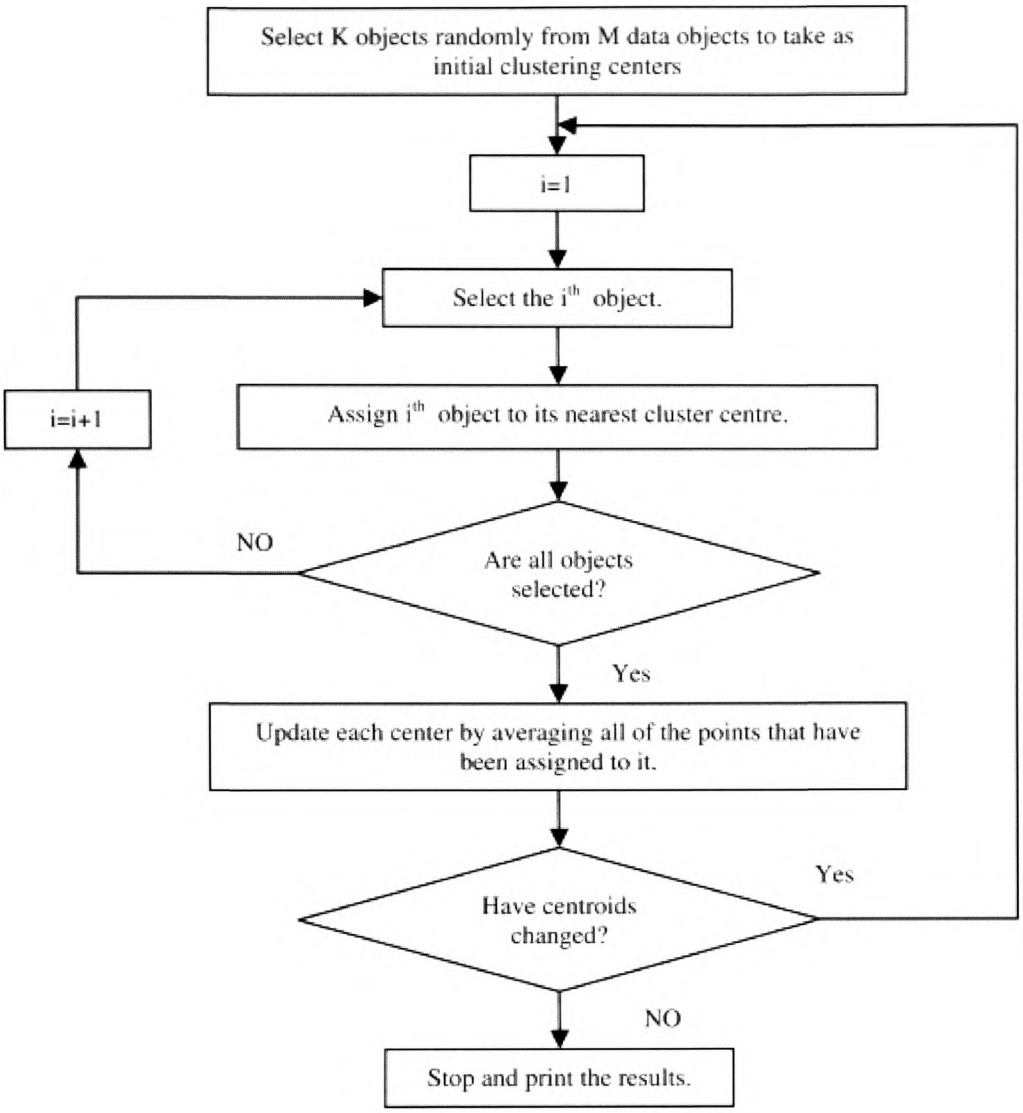

Figure 4-1 – Flowchart Summarising the Method Used by k-means Clustering

4.4. Canopy Clustering

Canopy clustering was another method of grouping data together which, compared to k-means clustering was faster by an order of magnitude in its execution. Canopy clustering initially separated the data into initial canopies using a similarity metric which was domain specific. The canopies were defined with two boundaries t_1 and t_2 as can be seen in Figure 4-2. These boundaries would be different depending upon the dataset used, and may take a few attempts to determine a suitable value for both variables [98]. The value of t_1 and t_2 would need to be determined depending upon the

actual data being analysed, and should be chosen with prior knowledge of how the expected clusters are described [97]. The data within the dataset would then be allocated to canopies depending upon the Euclidian distance from the point to the canopy centre.

The canopies were built one at a time with the initial canopy centre being determined as one of the points contained within the dataset as a starting position. The distance from this point to all others was then measured, if the distance between the canopy centre and the point being measured was less than the value of t_1 then it is said to belong to that canopy, and cannot be used as another canopy centre. Should the distance of the canopy centre to the measured value, fall within the t_2 boundary then the data point belongs to the canopy but may also start another canopy. This procedure was conducted until each of the data points within the data set were assigned to a canopy within the t_1 boundary [97]. This is shown in Figure 4-2, where each of the data points, had been covered by the t_1 canopy boundaries.

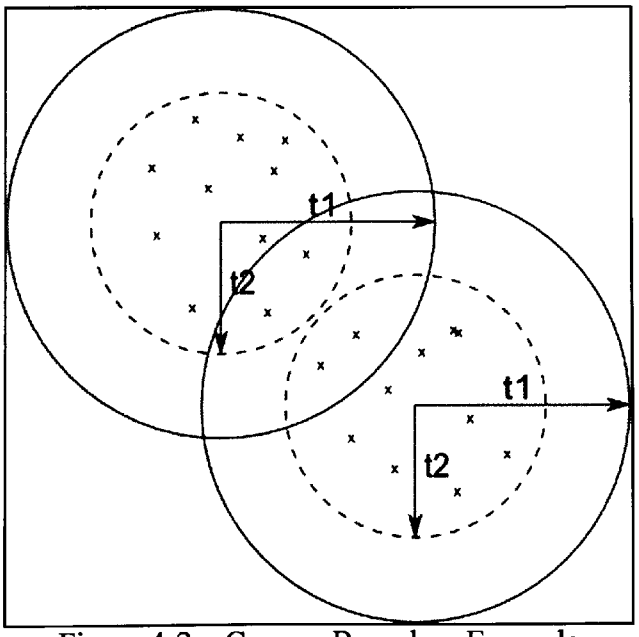

Figure 4-2 – Canopy Boundary Example

To further improve the fit of the initial canopies, the method of map-reduce was implemented. This not only enables a better fit for the data, but also enables the process to become highly parallel in its operation which speeded up the processing on large

data-sets [99]. The map-reduce function splits up the data into multiple datasets and performs canopy clustering on the individual datasets. The amount of individual datasets that are created can be determined depending upon the amount of computer processing cores that were available, meaning that the function was scalable to the hardware available. Once canopy clustering had been completed on each of the mappers, the canopy centres were sent to the reducer, which then performed canopy clustering on the centres and provides a list of new canopy centres.

The new canopy centres were then used for the final canopy clustering allocation, all of the values were then allocated to the canopies using the t_1 and t_2 boundaries. There could be some points that are not covered by the new canopy centres, these points have been allocated to the nearest canopy. The final stage of allocating data points to clusters was then completed.

The canopy boundaries were an important factor while conducting the canopy clustering process. The canopy boundaries should be chosen based on the expected cluster results of the data set, and this would vary depending upon the problem, and therefore some prior knowledge of the dataset was required [97]. By correctly choosing the boundaries of the canopies, the final solution should be covered by the canopies, and knowing this fact meant that when conducting further analysis within the overlapping canopies, only the data points contained within these canopies were considered. This canopy coverage of the final solution was what aided in the increase of speed within the algorithm, without the loss of accuracy in the final result.

4.5. Principle Component Analysis

Principle Component Analysis (PCA) is used to aid in the visualisation of multi-dimensional data in a two-dimensional space. The axis used for the first principle component is the one with the greatest variance and the orthogonal is taken for the principle component 2. Prior to finding the first principle component the data will need

to be standardized and cantered around zero [100]. The principle component can be found from a three-dimensional plot by rotating the plot till the greatest variance is found and the principle component axis will run through the zero of the axis with the use of least squares to determine the best fit of the data as shown in Figure 4-3 [100].

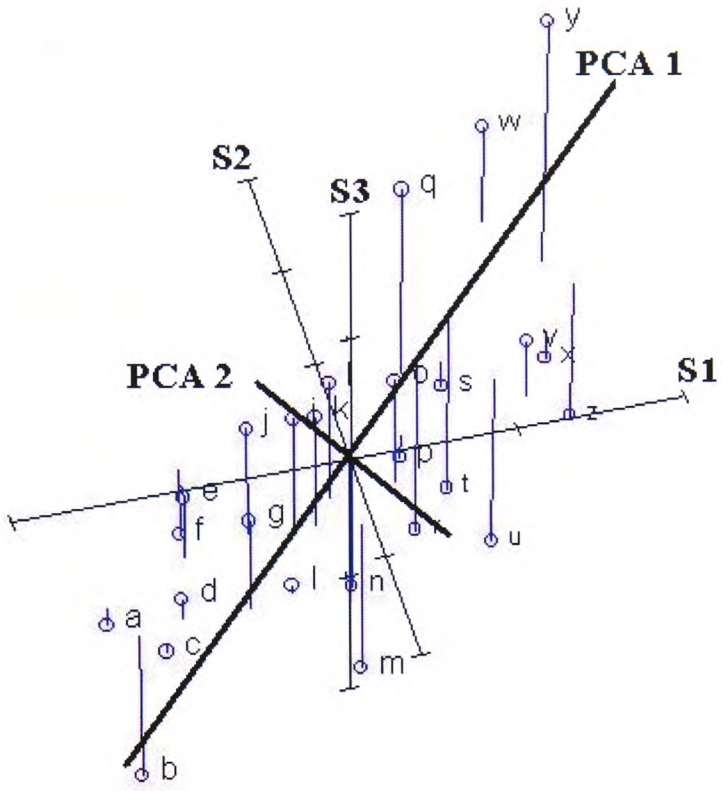

Figure 4-3 – Principle Component Analysis Plot

The data plotted onto the new axis can then be used for further clustering to determine the relationships in the data. With the use of NILM, the data points are required to be occupying a similar space within the axis, and therefore k-means clustering may be used to determine the final groups in the dataset.

4.6. Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is used for feature extraction before further classification of data. It is useful when there is high dimensionality in a dataset and aids

in the reduction of dimensions whist keeping the important information relevant to the problem being solved [101].

LDA works to maximise the the separation between classes so that they are more easily identified. When comparing LDA to PCA one of the main important distinguishing characteristics is that LDA keeps the variance of the classes the same, whereas PCA is developed around finding the lrgest variant for the first principle component. LDA also uses the mean of the dataset as the main discriminating factor unlike PCA which uses the variance of the data [101]. By using LDA the performance of classifiers is greatly improved and can be used in a variety of different clustering problems.

4.7. Hierarchical Clustering

Hierarchical clustering involved the grouping of data based on the Euclidian distance between points, the post popular form for displaying the clustering process was the use of a dendrogram. A dendrogram was a method of displaying the clustering data in a tree format and can be seen within Figure 4-4 which was created using sample data.

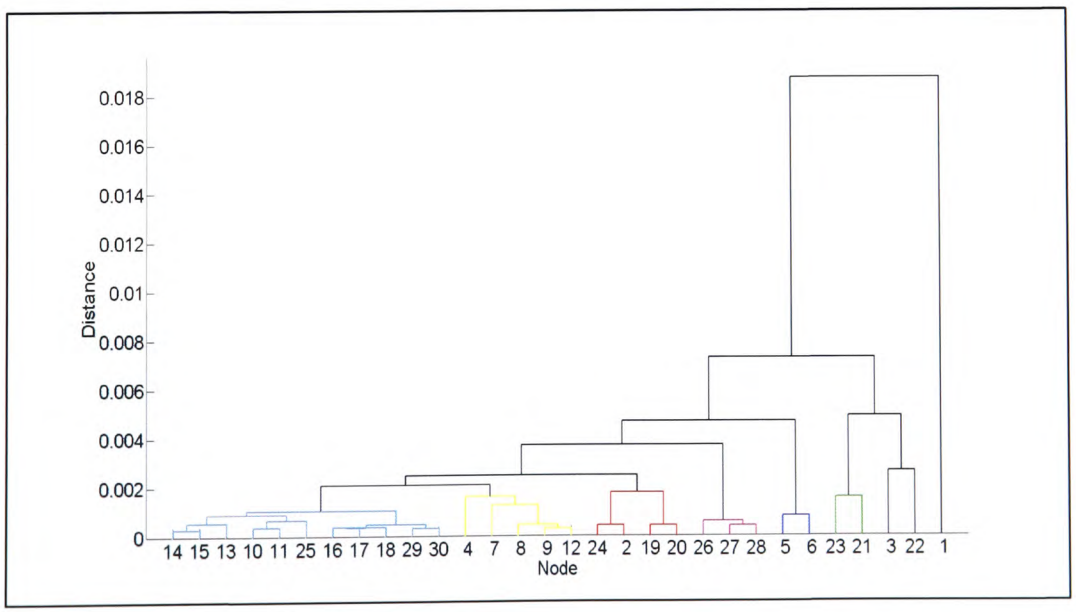

Figure 4-4 – Dendrogram Plot

To fully understand the dendrogram plot shown above, the basics need to be discussed. The dendrogram plot described the distances between the different nodes within the dataset. These distances were calculated using the Euclidian measurement, but other forms of distance metric may be used. The first stages of the algorithm join the two points within the dataset that have the shortest distance and the first connection is made. Further connections are then made, but when two points are connected, they are replaced with one point located mid-way along the line that connects the two nodes; this point is then used for all following calculations [102], [103]. The constant calculations of the distances between the data points, and the new points created when two data points are connected meant that the algorithm becomes computationally expensive the larger the dataset.

Figure 4-4 shows the different clusters that can be found within the dataset, and these are represented by the different colours. It should be noted at this point that the number of clusters that were found from the hierarchical clustering method is dependent upon the cut-off point that is determined by the user. The cut-off point Figure 4-4 shows clusters that are a minimum distance of 0.002 units. This can be tailored to the dataset, but some prior knowledge of the information being analysed is required.

4.8. Conclusion

This section has covered the concepts of clustering that could be used within the research. The clustering methods used will depend upon the application in which it is being used and the data that is being grouped. One important point to note is that each of the different methods requires inputs such as desired clusters or cut off points as an input to the algorithms, which are used to categorise the final groups of data. An example of this is the k-means algorithm, which required the amount of clusters that were to be found from the algorithm, which made this a difficult algorithm to apply to the problem if the desired number of clusters was not known.

K-means clustering was one of the methods that require the amount of clusters to be known prior to implementing the algorithm. This could be useful should the desired known outputs be declared, but in many cases this was unknown. Also k-means clustering was a recursive method of clustering and therefore could become computationally intensive.

Hierarchical takes a different approach to that of k-means, where the clusters are defined by the distance between the connecting data points. This could be identified by using the dendrogram plot, which plotted the distance between the groups, and the distance used for the segregation separates all of the clusters below this line into their own cluster groups.

PCA and LDA have been introduced, and are methods of reducing the dimensionality of the dataset, which is an inherent problem when dealing with NILM. By using LDA and PCA the dimensionality is reduced whilst still containing the relevant and important information needed to solve the problems of classification, but require the data to be manipulated in a way to so that the features are chosen to provide the maximum variance in PCA or the maximum class separation in LDA.

This chapter has also introduced the idea of canopy clustering, which unlike Hierarchical and k-means does not require a cut-off or a known number of clusters to be previously known. This provided an advantage when the number of clusters was unknown due to the properties of the clustering algorithm. The size of the canopies was chosen depending upon the dataset, and as long as the canopy boundaries were selected to be large enough to contain the final results of the clustering process, the number of clusters prior to computation was not needed. Data was segregated into overlapping canopies, which were then further analysed to determine the final cluster memberships which greatly increases computation time, but was also able to deal with large numbers of datasets and clusters. Canopy clustering still required the use of further

computationally intensive clustering methods such as k-means, to describe the final cluster membership. The k-means clustering was conducted post canopy clustering, and was only computed on the data points contained within the canopies, and all other canopies are ignored.

By using off site services to conduct clustering for NILM, the power of parallel processing could be harnessed, which could be used for the map-reduce functions within the canopy clustering algorithm. This part of the processing is only completed at the time of training meaning that the pressure on the system is only present during the training period.

Chapter 5. Methodology

5.1. Introduction

This chapter shows the research methodology used within the research. The methods for the primary research have been developed form the findings of the secondary research and from discussions on how a NILM system could be implemented in a real world environment.

5.2. Research Methodology

The research methodology followed a classical engineering approach consisting of both primary and secondary research. The secondary research undertaken was in the form of an extensive literature survey that provided support to the direction of the work undertaken.

As the thrust of the research was to provide additional knowledge in both theoretical and practical domains, this secondary research was built upon using the strategy of developing the work through simulations, with validation of the results by experimentation. The literature survey was comprised of peer reviewed academic literature and a survey of historical and contemporary industry procedures.

The strategy for the primary research necessitated the creation of an experimental infrastructure. This consisted of a data gathering architecture that was calibrated to ensure that consistent and valid data was acquired. Data hardware specifications were created to capture the data, and used in conjunction with the required development software. The data capture specification was considered and the final sampling rate and resolution of the analogue to digital converters was decided upon based on the design requirements stated in Chapter 6, to ensure that a good representation of the original signals was replicated.

Hardware specifications for the sensors used within the research were decided upon for the acquisition of the current signals. The sensors were chosen from the results of the technology review, and their relevance to the final usage with a fully implemented system. The decisions were based on accuracy of the sensors, cost and ease of retro-fit installation of the sensors in the relevant environment in which they would be used.

The data recorded had to be pre-processed to convert it into the correct representation that the final analysis was conducted on as described in Chapter 7. Software specifications were considered for the pre-processing of the data, which required the ability to import and manipulate the data, and also perform the required signal processing procedures on the data which has been considered in Chapter 3. To ensure that the data conversion was robust and representative of the actual signals being recorded, digital filters were employed to remove potential sources of errors.

The groups of loads contained in the total current draw needed to be identified, and to be able to do this each of the individual loads were recorded as well as the total load current. This allows for the monitoring of the individual loads so that the component parts of the total load could be defined, and the final groups that require definition from NILM process could be conducted. The idea that the groups of loads would be initially clustered rather than trying to determine the individual loads from the outset was decided upon due to the final usage of the system in the real world. When considering how appliances and loads were utilised within domestic premises, they would be in operation in parallel, and therefore the loads will be grouped into a single cluster of data for the total current draw.

To ensure the validity of the results, the filtered current signals for the individual loads required further analysis to ascertain their operating state. Each of the loads was processed using the analysis software where the operating conditions of the loads were

determined and given a code that identified the operating state of the load, and described to ensure that the final results could be compared to the theoretical outcomes.

The individual loads were then combined to create the final groups to be identified from the NILM algorithm. This was completed by combining the load identification codes from each of the individual loads, and allows for the total loads to be identified as a composition of the loads operating conditions.

Canopy clustering was used for the initial separation of the data due to the benefits outlined in the clustering review. It provided a fast an efficient method of initially segregating clusters within a dataset, which were plotted in a two-dimensional space. By using a measurement of similarity, which in the case of NILM was the value of the currents within the specific domains, the clusters could be separated out into overlapping canopies that cover the expected final clustering result. This negated the need for more expensive clustering methods to be used such as k-means clustering, which required further information regarding the dataset such as the number of anticipated clusters, which is unknown.

The algorithm development for the canopy clustering model was completed using known datasets which were recorded and individually analysed. The relationships between the boundaries and canopy centres were found from the known loads within the dataset. Different relationships were analysed for the different types of loads that could be found within a typical implementation of NILM within a domestic environment, which included resistive and non-linear loads. The relationships were analysed to ensure that an overall approach to the canopy clustering algorithm could be found, as there was not the possibility to pre-determine different algorithms depending upon the load type due to the nature of the process of NILM.

With the algorithm developed for the initial canopy clustering of the data, further measures had to be put into place to ensure that the best fit for the data was

found. The initial canopies were created using the random functions provided within MATLAB. There were more canopies within the dataset than were needed to cover the whole clusters contained within the data which led to a large amount of overlapping canopies. To be able to get a better fit and reduce the number of canopies within the results, map reduce methods were used. This method was chosen due to its ability to produce better results for the canopy clustering algorithm which were found from the literature survey.

Map reduce was used to remove the redundant canopies, by conducting canopy clustering on the centres of the canopies found in the previous stage of the algorithm. This removed excess canopy centres and produced a better fit for the data. Map reduce also had the benefit of being processed in a parallel fashion, and therefore is scalable to the amount of processing power in the system in which the algorithm was being processed.

The results from the map reduce were then applied to the data set to assign each of the points to its canopy which it is covered by. The map reduce methods moved and combined the canopies for a better fit, and as a result of this there were some points that were not covered. This had to be rectified as each of the points had to belong to a canopy, and therefore the stray data points were allocated to the nearest canopy centre, with the Euclidean distance being the unit of measure, which meant that the dataset was fully allocated to the clusters. Actual stray points within the dataset caused by transitions of turning on and off loads could be seen separate to the clusters and made up canopies of one point alone, due to the points being found in between the clusters.

The canopy clusters were created to cover the clusters within the dataset, but the profiles needed to be created for the clusters covered by the canopies. To be able to create a profile of the groups of loads, further analysis was required on the canopied data, to identify the actual clusters. There were some canopies within the dataset that

were overlapping, and contained two separate clusters, which had to be separated out. The required further computation for separating out the two clusters and this is where more computationally intensive clustering algorithms were employed such as k-means clustering.

To be able to separate out the clusters within overlapping canopies, k-means clustering was used. The initial problems of using k-means clustering over all have been alleviated, as the dataset has been separated out into subsets of data defined by the canopies. This negated any need for distance measurements to be calculated between points in the canopy to those outside of the canopy, as those data points that reside out of the canopy were deemed to be too far away and therefore could not be a part of the final cluster.

The clustering process that was completed within the canopies meant that profiles could be created for the groups of loads that were found within the dataset. These profiles described the operating parameters of the group of loads or the individual load that was present. The profiles created needed to show that the final profiles were representative of the expected loads as found from the pre-processing stages of the research.

The results were evaluated through the comparison of the model clustering output to the expected results for the different canopy models that were created. Each of the models was tested and the best one used for further testing on different loads to validate the final results of the canopy clustering model. Real world examples of loads were used for both the canopy model creation and the validation of the canopy model.

Once the canopy model had been created, the analysis of the way in which loads interact was determined. The canopy clustering models were built on the proviso that the canopy clustering model would be able to separate out groups of loads in the first stages, which was how real world loads will be seen as was found from the secondary

research. The loads that were used within the canopy model creation and testing were further analysed to show the relationship as to how the loads could be added using trigonometric identities. This was done so that the groups of loads found from the canopy clustering could be reversed to find out the component loads that make up the group of loads found. This form of load disaggregation was decided upon as the loads will have profiles associated to them, which could then be called upon to disaggregate the final loads used within the premises, and the energy values and the time of use of the loads could then be recorded for further consumer feedback.

5.3. Conclusions

The chapter has detailed a working research methodology that has successfully shown that development of the research ideas into the final algorithms used has been successful. Methodical experimentation procedures have been developed on a design, analysis and validation approach which have aided in a robust and repeatable experimentation procedure. This approach provides for a robust and high quality research undertaking.

Chapter 6. Experimentation Procedure – Data Capture

6.1. Introduction

This chapter provides a review of the approach to the empirical processes undertaken as part of the research. Explanations of how the experimentation has been created from the information obtained within the literature review and developed from the methodologies in chapter 4 are detailed. The chapter explains the set-up of the hardware, and the programming of the software for the capture of the data, with the view of satisfying the design criteria of the experiments.

6.2. Hardware Considerations

The hardware used within the research was required to obtain empirical data that would be explored and analysed within the NILM domain. The technology review has brought together the technologies that were considered, and explained which technologies provided the best methods of capturing the data.

The experiments were set up to capture raw data that could be used within further analysis tools such as MATLAB. The experimentation design considerations required that the current draw of each of the individual loads should be recorded with the current and the voltage of the total load also being recorded. For the sensing of the currents a current transformer was used, with a rating of circa 50A which is adequate for the research being conducted, but when considering implementation within the domestic environment, a current transformer rating of 80A is required. A voltage transformer was selected to step the voltage down to the correct levels as used by the DAQ hardware.

The current transformers were used to convert the current signals into voltage, and this was done by connecting a resistor to the ends of the coil, so that the voltage of the output was directly related to the current at the input. The actual voltage ratio for the

output was described as the ratio of the turns within the coil (1/2500), and the value of the resistor on the output (68Ω):

$$Vout = \frac{I}{N} \times R = I \times 0.0272 \quad {}^{V}/_{A} \qquad (6\text{-}1)$$

The voltage outputs were connected to the PCI-Base and the actual devices was controlled through a C program, and the provided API was used for setting up each of the channels with the various channel specific information such as the analogue to digital conversion range which in this instance is -1V to 1V and the sampling frequency of 4000 samples per second per channel. The code for the recording the data and setting up the DAQ device can be found in Appendix A – DataCapture PCIBase II.

For the use of NILM, contemporary research has shown that the initial harmonics are the most important and contain the most relevant information [43], [55], also as the number of harmonics to be analysed is increased, the higher frequency harmonics become lost in the noise. Determining the number of harmonics that were required for the analysis depended upon the type of loads that were being monitored. Considering a purely domestic environment, there are many resistive loads such as kettles and toaster, but also non-linear loads which require further harmonics for analysis, Srinivasan et al [43] showed that that information contained within the first 16 harmonics could be used for classification of loads within the domestic settings. In theory this would work, but the information contained within the later end of the harmonic spectrum becomes small and there could be issues with the signals being lost within the noise of the system.

From this information a sample rate of 4000 samples per second per channel was chosen. This rate of sampling was used within the acquisition program to capture the data, and the rate of 4000 enabled a final frequency analysis of up to 450Hz to be

monitored without aliasing problems, giving a total of nine harmonics that could be used for classification.

A channel was used for each of the loads that were being monitored, and also the total load. These samples were recorded in their converted from using (**6-1**, and the actual current being consumed was written to file. The files were organised using comma separation (CSV) with each line within the file representing one seconds worth of data captured, which made the next stage of processing simpler within MATLAB. The channels were set up with the API using the commands described in Code Sample 6-1.

```
chav[1].cha = AD_CHA_TYPE_ANALOG_IN|2;
chav[1].store = AD_STORE_DISCRETE;
chav[1].ratio = 1;
chav[1].trg_mode = AD_TRG_NONE;
chav[1].range = 0;
```

Code Sample 6-1 – Channel Setup

This describes how the channel had been set up and was completed for each of the channels. The 'chav' variable was a structure defined by the API for the description of the channel, and an array of the structures had been created to contain the information of each of the channels being monitored. The variable 'chav[1].cha' was the channel type and location and in this example it described an analogue channel in, on channel 2. The 'chav[1].store' was the description of how the signal was stored, where the value stored was shown to be a discrete value. The 'chav[1].ratio' of the signal defines which samples were to be stored, and in this case each sample was stored. The recording from the channels was completed as soon as the program was started, and therefore there was no need for a trigger function to be included. The trigger function was set to off using the 'chav[1].trg_mode' variable. Finally the range of the signal was set up using the 'chav[1].range' variable. In this instance the range was set to zero which represents a range of ±1.024V.

Once the channels had been set up with the above information, the actual scan information needed to be set as shown in Code Sample 6-2. This information as shown below was used to describe the actual acquisition of the data, and how much data should be obtained.

```
sd.sample_rate = 2.5e-4f;
sd.prehist = 0;
sd.posthist = 22118400e58;
sd.bytes_per_run = 4000;
sd.ticks_per_run = 4000;
```

Code Sample 6-2 – Sample Information

The 'sd.sample_rate' described the rate in which the samples were taken, and here represents the acquisition of 2048 samples per second for each of the channels. The 'sd.prehist' was used when triggering functions were used, and will be used to record a user defined number of samples that occur before the trigger has happened. Within this experiment there was no triggering setting, and therefore the 'sd.prehist' value was set to zero. The 'sd.posthist' represented the total amount of samples that were taken. The 'sd.bytes_per_run' and 'sd.ticks_per_run' were used to represent how many samples were recorded per channel.

The API record the data in runs, and these runs were made up of all the channels together, so for this, a variable was created for the total number of samples that were taken for the complete run of all the channels recording 2048 samples each. The start of the data acquisition was initiated with the command shown in Code Sample 6-3.

```
rc = ad_start_scan(adh, &sd, 5, chav);
```

Code Sample 6-3 – Start Scan Command

The 'rc' variable was used as a return function call for the start function, and used for error checking. The attributes that have been passed to the function include:

- 'adh' – which is a reference to the device that is being used for the data capture, and has been initialised at the start of the program.

- '&sd' – The address of the structure for the sample information such as sample rate and other variables that have been described earlier.
- '5' – This was the number of channels that are being monitored, so will change depending upon the required number of channels to be monitored.
- 'chav' – The channel description structure which had also been described previously.

The previous function was used for the start of the sampling process, but the actual data acquisition was completed in runs, and therefore each run needed to be initiated using the following command, and are described in Code Sample 6-4.

```
rc = ad_get_next_run_f(adh, &state, &run_id, samples);
```

Code Sample 6-4 – Continue Sampling Command

The function was requesting the next run required additional variables that described how the next run was conducted. Firstly there was the '&state' variable, which was used to monitor the operating condition of the program, and if there was a fault within the recording process. The second variable of importance was the '&run_id'. This variable recorded the state of the program that showed the actual run that was being completed, and therefore could be used as a measure as to how far through the recording process had proceeded. The 'samples' variable was included in the function call, which was the variable used for the recording of the data, and was used to hold all the samples from all of the channels for each run that happened.

Once the run had been completed, the information within the 'sample' variable needed to be processed so that each of the channels could be separated, and written to their respective files for further analysis. The API for recording the data writes the samples to the variable in blocks, where in this program each channel being recorded per run is made up of 4000 samples. The 'samples' variable was split up into adjacent blocks of 4000 samples, with each block representing each of the channels. The samples

recorded within the variable were the voltages recorded across the resistor terminating the current transformer, which was calculated back to the current value using equation (**6-2**, where the value 2500 was the turns ratio for converting the secondary current to the primary current:

$$I = \frac{V}{R} \times 2500 \quad \text{A} \tag{6-2}$$

To ensure accurate results have been obtained, the resistor values were measured in a constant room temperature, and those values were used within the calculations for determining the input current being measured. The program excerpt Code Sample 6-5 shows the code used for calculating the current and writing the value to a file.

```
for (int i=0; i<4000; i++)
{
        fV << samples[i]*894 << ", ";
        fIT << samples[i+4000]*36.6569 << " ,";
        fI1 << samples[i+8000]*36.6569 << " ,";
        fI2 << samples[i+12000]*36.5497 << " ,";
        fI3 << samples[i+16000]*36.6569 << " ,";
}
```

Code Sample 6-5 – Write Sample to File

Once all of the samples had been completed the program was terminated with the 'ad_stop_scan' command. This ensured that the hardware recording had been terminated correctly, and ready for use with further testing. The command can be seen in Code Sample 6-6.

```
rc = ad_stop_scan(adh, &scan_result);
```

Code Sample 6-6 – Stop Sampling Command

6.3. Calibration

For the system to be truly representative of the actual loads that are being measured, the system needed calibration. The calibration was carried out using the meter testing equipment supplied by KIGG Ltd. The correction factors for the current and voltage transformers were used, as shown in Code Sample 6-5, during the sampling process when the conversion from the voltage sample was converted into the ampere reading, with the same being completed for the voltage readings .

The current transformers used were for the measurement of the current, and due to the set-up of the system where the transitional points of a load being turned on were not used for the analysis, the transient analysis of the current transformer was not considered to be used within the analysis, whereas the steady state analysis was. Any phase change in the current measurements would be considered during the analysis stages when the phase of the current reading would be compared to that of the voltage reading to find the phase angle of the device.

6.4. Conclusions

The chapter has introduced the data capture procedure of the research, and how the use of pre-validated products such as the PCIBase had been used to capture data for further analysis. The integration of the current transformers and the high resolution of the PCIBase meant that accurate representation of the input current had been obtained, and through this the signals can now be processed within MATLAB to build up a NILM system around the captured data.

The data capture set up had been completed to ensure that all relevant information was obtained, which included recording the current signals from not just the total current, but all the individual loads respectively so that accuracy of the results could be validated.

Calibration of the equipment was required to ensure that the results were accurate and representative of the loads that were being monitored. Steps were taken to ensure that the correct scaling factors were used to convert the voltage samples to the current readings during the sampling process.

Chapter 7. Data Pre-processing

7.1. Introduction

This chapter describes how the data taken from the data acquisition device, as described in chapter 5, and converted into useable data for further analysis within the NILM process. The section starts by introducing the importing of the data into MATLAB which was the chosen program for analysis within the research, and how the data was converted into its final usable datasets that could be used for the exploration and creation of models and profiles of loads that could be used within NILM for classification.

7.2. Data Pre-processing Overview

The overview of the data pre-processing flow chart can be seen in Figure 7-1. Each of the processing has been described in detail within the following sections.

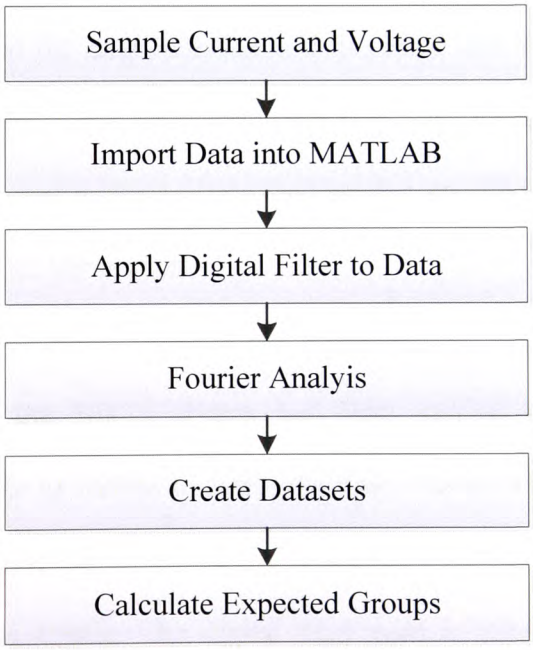

Figure 7-1 – Data Pre-processing Overview Chart

7.3. Data Import

Through the programme of work the use of harmonic signatures had been explored, and was used within the research as the classification data for the NILM system. The raw

data was imported into MATLAB using the inbuilt import functions. The data acquisition program had been developed so that the data format of the raw data was compatible with the MATLAB functions.

Each of the individual channels were imported into the workspace and given their respective variable with a number suffix to identify the individual loads, and also the total load current is imported into the workspace. Each of the variables that were imported into the workspace were represented by a two-dimensional array where each row consisted of 4000 samples, which is equivalent to one second within the time domain. A sample period of one second was used due to the considerations of how domestic load monitoring would be implemented, and a sample resolution of one second would be sufficient to monitor the activities, without the loss of data.

7.4. Data Filtering

The data within the variables required filtering as there were no hardware filter applied to the input signal, and the data recorded and imported into MATLAB was unfiltered and may have contained harmonic components that were not required for the analysis, so higher frequencies were removed from the signal to ensure the results of NILM were not skewed by the higher harmonics.

The filter used within the system was a digital third order Butterworth filter. The analysis used within the NILM system had been carried out using the frequency components from 50Hz to 150Hz in 50Hz multiples but provisions were built into the system so that up to nine harmonics could be used, and therefore the cut off frequency for the filter used was 450Hz. The digital filter used is inbuilt within the MATLAB Signal Processing Toolbox, and the transfer function variables were obtained using the Butter function [104], and can be seen in equation (7-1).

$$H(z) = \frac{0.0317 + 0.0951z^{-1} + 0.0095z^{-2} + 0.0317z^{-3}}{1 - 1.4590z^{-1} + 0.9104z^{-2} - 0.1978z^{-3}} \quad (7\text{-}1)$$

Different filtering technologies were considered, such as active and passive filters [105], [106], [107], but due to the nature of the research, and the implementation of the final proposed system within a domestic environment, the active and passive filters require hardware configuration, whereas the digital filtering can be completed at time of analysing the results making the whole process from installation to analysis of the signals more streamlined.

Using a third order Butterworth filter allowed the wanted frequencies to be passed through, and the nature of the Butterworth meant that there magnitude response was flat across the pass frequencies, and the third order allowed a sharper roll off towards the stop frequencies at a rate of -60dB per decade [108]. The bode plot of the filter are shown below Figure 7-2.

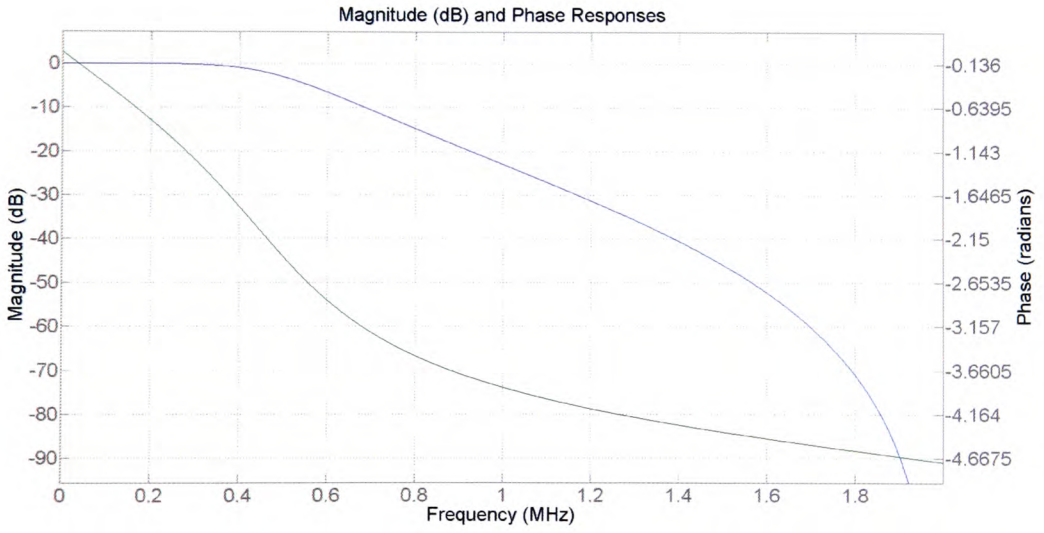

Figure 7-2 – Phase and Frequency response of Third Order Butterworth Filter

7.5. Pre-Processing and Fourier analysis

Each of the rows of data needed to be converted to display the harmonic information of the signal on a second by second basis, which was completed by using Fourier Analysis

on the data. The inbuilt Fast Fourier Transform function was used to convert the current time signals from the time domain to the frequency domain. The FFT function outputs the real and imaginary coordinates into the variable, which could then be converted into the magnitude only of the actual harmonics.

Before converting the complex coordinates to the magnitude only component, the RMS value of the harmonic was required to be calculated, and therefore the output of the FFT function was divided by $\sqrt{2}$, and the output was also divided by half of the length of the sample size as in-keeping with the FFT Algorithm. To calculate the magnitude of the harmonics, the inbuilt 'abs' function was used, which basically converts the complex coordinates into the magnitude only representation.

Each of the current signals were converted into the frequency domain, and due to the nature of the FFT algorithm, the frequency spectrum was mirrored on both sides, and therefore only the first $\frac{N}{2} + 1$ samples were taken into consideration for the analysis within NILM. Below shows the initial code for importing the data from its original format and converting it into the frequency domain with the RMS values used shown in Code Sample 7-1.

```
I1=importdata('I1.dat');
s=size(I1, 1);
I1 = fft(I1)./(2000*sqrt(2));
I1 = I1(1:s,1:2001);
I1abs=abs(I1);
I1angle=angle(I1);
```

Code Sample 7-1 – Data Import and FFT

The phase information of the system was contained within the FFT output and needed to be extracted. This was completed by using the 'angle' function which basically converts the complex numbers into the angle via equation (7-2), using the imaginary and real components of the FFT solution.

$$\beta = \arctan\left(\frac{imag}{real}\right) \qquad (7\text{-}2)$$

The phase alone for the current signals was not enough as for it to have any value in the context of the problem, the phase required a reference. The reference phase used was contained within the voltage variable of the data, and to obtain the actual phase shift of the current with regards to the voltage the phase of the voltage was calculated using (7-2, and the difference of the phases was calculated. This difference in phase was then stored within a variable in the workspace and became an important component used when evaluating the make-up of the groups of loads that needed to be classified. The formula for the actual phase shift with respect to the voltage is shown in (7-3).

$$I1_{phase}=V_{ang}-I1_{ang} \qquad (7\text{-}3)$$

This was calculated for each of the currents that were within the dataset, along with the total load current. By storing this information the signals that were being analysed could be rebuilt using the phase and magnitude, and facilitated theoretical checks on the results that had been obtained.

7.6. Data Organisation

The previous section described how the data was converted from the time domain to the frequency domain, and how the output of the data was contained in a large array with the harmonics being placed at every 50Hz interval. Due to only the harmonics being important, there was a requirement to remove the excess noise, and therefore datasets have been created for each of the individual and the total load currents.

MATLAB provided a function for creating datasets, and is basically the equivalent of setting up a structure similar to C/C++. This made data flow easier to follow, and also aided with the monitoring of the different variables within the system.

7.7. Considering Load Operating Conditions

When considering NILM as a concept, to be able to correctly identify the different loads that were present within the system, each of the individual loads was required to be pre-processed so that the operating conditions of each of the loads could be identified. For the purpose of the research, analysis of the individual loads was carried out, and from this, the actual status of the NILM systems could be defined at any time during the test period. However when looking a practical implementation of NILM, this was not viable, and this process had been completed to validate the results only, thus there was a requirement for load profiles.

To be able to identify the operating conditions of each of the loads, and therefore the operating status of the total system, the individual loads were analysed within the frequency spectrum and within a two-dimensional space which is composed of the fundamental signal and third harmonic. When looking at the loads in this two-dimensional space, where the fundamental and third harmonic were used for the variables, clear groups could be seen, which required to be separated, and the groups numbered for each of the loads. The numbering of the groups was completed to track the operating conditions of the individual loads, and the combination of the individual loads was used to compute the overall group at each instance of time on a 1Hz Cycle.

The code for the algorithm for defining the groups that are present is shown in Appendix B – Load Operating Conditions Code with the flow chart describing the procedure shown in Figure 7-3.

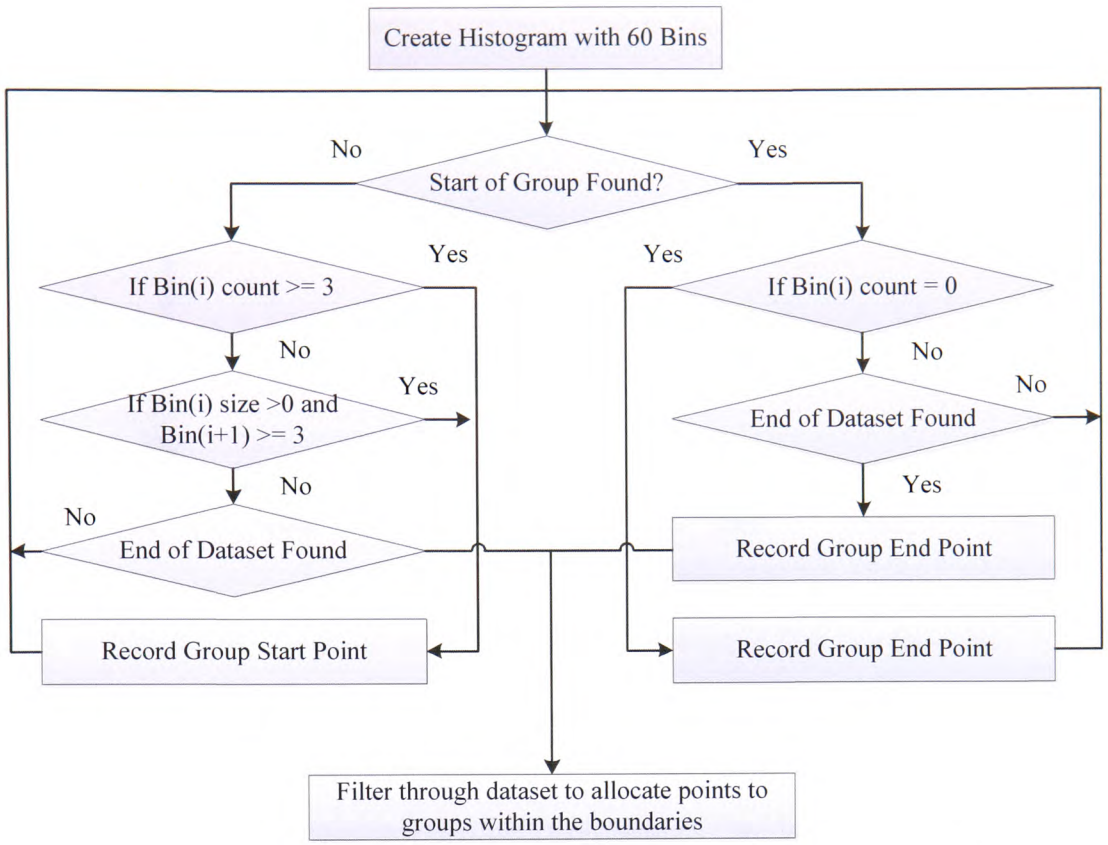

Figure 7-3 – Load Operating Conditions Definition Flowchart

To fully understand the operating characteristics of the loads, the use of histograms to display energy consumption of various loads was implemented. By plotting the samples into bins within a histogram plot, a clear distinction could be seen between the different operating conditions, and when the load was in its off state. As an example the results from a toothbrush charger can be seen below in Figure 7-4.

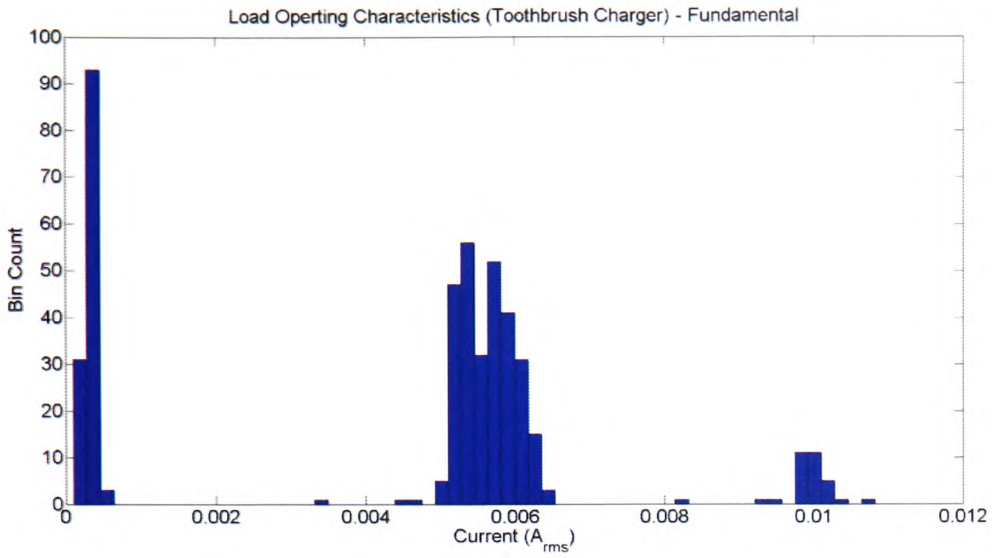

Figure 7-4 – Load Operating Conditions Histogram Plot Fundamental

This example shows that within the fundamental frequency there were 3 clearly identifiable groups that were required for classification. The first group was situated on the far left of the plot around $0A_{rms}$ and this group indicates the off position of the load, and hence is consuming no current. The other two groups centred on $0.006A_{rms}$ and $0.01A_{rms}$ are the two on operating conditions that were identified as two separate entities which could be described as the initial current charge when the toothbrush was plugged in, and the steady state charge which was its normal charging cycle.

Just using the fundamental plot alone does not give the whole story, as there could be two identifiable groups represented as one, but that was only apparent when looking into the higher harmonics. This was where the 3rd harmonic was taken into consideration and the same procedure was applied, with the result shown in Figure 7-5. Here the groups were less defined, but there were still clear gaps between the groups where the bins were empty, and this indicated the separation of the groups, and could thus be used for determining the overall composition of the load operating conditions.

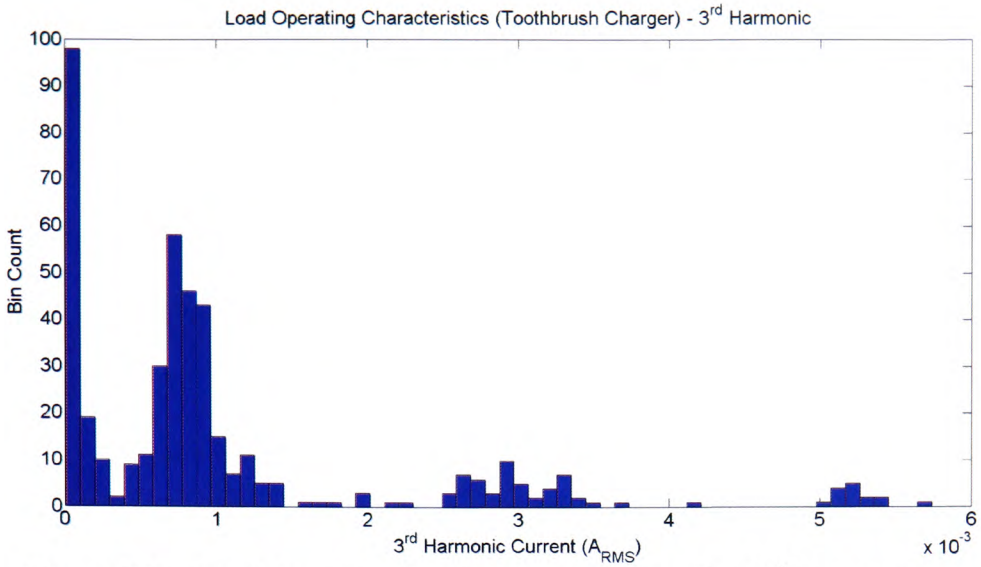

Figure 7-5 – Load Operating Conditions Histogram Plot 3rd Harmonic

When considering both the fundamental and 3rd harmonic in the two dimensional space, the groups could be clearly separated using the boundaries found within the previous stages. To complete the process of allocating data points to the operating condition of the individual load, the points were allocated to the group depending upon which boundaries the points fall between.

It should be noted at this point that there were some points that did not belong to any particular group, and can be seen, both within the histogram plots as well as the scatter plots, to be outliers. These points were present due to the loads change in operating conditions, and represent a transitional point. These transitional points were not going to be repeatable, as the point at which it is found will depend on where in the sampling period that the load was turned on, and therefore in this scheme provide no significant importance to the work.

To be able to correctly identify the groups present within the dataset, an algorithm was developed to differentiate the individual group by monitoring the bins, and combining the adjacent bins that have more than one data point within it. This process filters through the whole dataset just looking at the fundamental and third

harmonic, and outputs were stored within a variable for the individual components for each of the loads.

When considering the dataset, there were a few key concepts that need to be taken into consideration when looking at defining the groups, and there were a few assumptions that needed to be made about the group characteristics. When deciding upon making a group from the histogram plot, the number of data points within the bin was assumed to have three or more points within it, anything less, and this was considered as a transitional point. However, when there were adjacent bins containing one single data point within the first bin and more than three data points in the next adjacent bin, then this may be considered the starting point of the load operating parameters. The end of the group could be found once a bin was being searched and there were no data points within the bin, or if the end of the dataset had been reached. The end of dataset check has been included as the end of a group may have terminated at the final bin allocation.

Once the creation of the groups was completed, each of the data points within the individual loads was sorted into their respective groups. This was a process of checking to see if the data points were within the group boundaries within the fundamental and third harmonic, and was allocated to the groups within both the fundamental and third harmonic, with the transitional points being removed from the analysis.

The operating characteristics of the individual loads have been obtained, and were represented in the fundamental and third harmonic, and the consolidation of the groups within the two domains was carried out. This analysis gives the boundaries within a two-dimensional space in which data could be allocated, and each individual data point within the dataset is allocated to a particular group depending where the point is within the space and the final allocation can be seen in Figure 7-6.

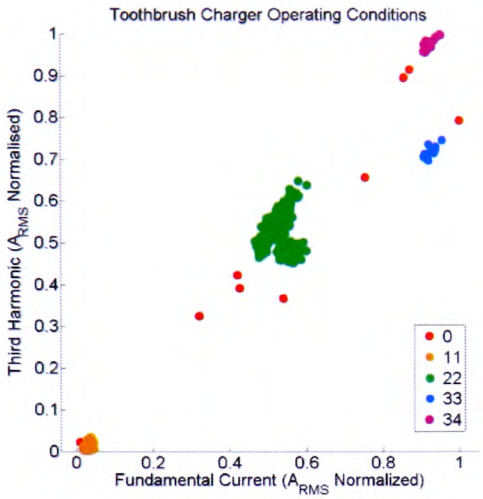

Figure 7-6 – Toothbrush Charger Scatter Plot of Groups

To be able to correctly identify the groups that were seen in the load, a variable described the load characteristics with four digits in two parts as seen in equation (7-4). Here the first two digits represent the group allocation within the first load, and the final two digits represent the group allocation in the second load. The group 1 load is multiplied by one hundred to move the group by two decimal places, so that the addition of the second group could be added. A final representation of the group ID's can then be determined by removing the duplicate groups from Group Total, and a profile of the group can be built including minimum, maximum and average values for the group.

$$GroupTotal = (Group1 \times 100) + Group2 \tag{7-4}$$

To be able to correctly use canopy clustering within the research, it was important to gain information about the clusters which included the ratio of the radii of the group to the value of the centre of the group. This ratio could be used when

determining the values of T_1 and T_2, and therefore the ratio was calculated for each of the individual groups and added to the list of characteristics of the loads.

Other important characteristics of the loads that were a requirement for the correct classification of the loads from within the groups was the phase shift of the current. The phase of the current was going to be important, as the interaction with other loads would determine the final total current that was being drawn.

The way in which the currents interact with different phases required careful consideration, and a look at the fundamentals of Kirchhoff's Current Law (KCL) and some trigonometric identities. KCL states that the currents entering an electrical node within a circuit equal the current leaving the node, and therefore from this, it can be said that the total current recorded at the point of entry to the premises is equal to the total of the currents drawn by the loads within the premises.

When FFT analysis was conducted on the total loads, the sum of all the loads would be present within the frequency domain, and as such each of the frequency components would be equivalent to the combination of the individual loads frequency components. Therefore it was important that each of the individual load profiles were created with the relevant phase shift and amplitude, so that the trigonometry identities may be applied.

7.8. Analysing Values of T_1 and T_2

The values of $T_1$and T_2 were found for the individual loads and groups of loads as described in section 7.7. When considering these variables, there was an issue with finding a fit for the variables that can be used across the many different loads. The reason for this was that depending upon the load being analysed, there were varying values for T_1 and T_2 depending upon the value of the fundamental current draw. To alleviate this issue the values of the radius of the different groups was analysed, and using the tools within MATLAB such as the curve fitting toolbox, values of T_1 and T_2

could be determined from the relationship between the radius of a group and its centre fundamental value.

7.9. Conclusions

To successfully implement NILM, the process of capturing and converting the data into useable information was the most important part of the process. It was vitally important that the data being used to model the load analysis on is accurate in its representation of the real world equivalent.

By using MATLAB for the data import and analysis, the full process of the data management and organisation could be followed to ensure that each step of the processes was executed correctly. MATLAB allows the data to be stored in the desired way, and the use of datasets allowed for the easy management of the data.

One important consideration for using MATLAB was its extensive use of toolboxes that added extra functionality to the program. The use of the Signal Processing Toolbox facilitated the development of digital filters that were used within the system. By adding the ability to filter the data, the risk of high frequency aliasing was reduced due to the removal of the high frequency harmonics by the filter. This allowed the accurate models to be developed from the real world datasets, without the danger of high frequency noise skewing the results. The filter implemented within MATLAB has shown to reduce the high frequency components, but one factor that needed to be taken into consideration when analysing the results from the filter is the phase shift of the different frequencies; this is why a zero phase shift filter was implemented so that the phase of the loads was not affected.

Using the built in Fourier analysis tools, the conversion of the time domain signal into the frequency domain was completed, which eliminated any margin for error in the programming of the conversion. One note that was considered from the Fourier

analysis was that the outputs for the magnitude of the signal were given as the peak value, and the conversion to RMS can be computed using the RMS conversion factors.

One issue found with using the Fourier analysis was the phase shift of the different frequencies. The angle information taken from the output of the Fourier transfer could not be used directly, as its information was useless on its own. Only when taking into account a reference for the phase shift did the information become of value. The current phase shift should be taken into consideration when comparing it to the phase of the voltage, and this would provide the actual phase shift of the current.

The chapter has also covered the importance of monitoring the individual loads within the scope of developing a model for NILM. By looking at the individual loads, an overall picture of how the total groups would interact could be found, and a final picture of the total load could be clearly seen.

The monitoring of the individual loads has shown that the loads will display different operating conditions, which complicates the processing of the information within NILM. No longer can loads be considered in an on/off state, but there will be intermediary states also which would require consideration during the classification process. This part of the research was important from the view point that the final classification of the loads could be validated as the load profiles of the individual loads were known.

Chapter 8. Canopy Clustering Algorithm

8.1. Introduction

The canopy clustering algorithm adopted within the research is described within this chapter. It begins with the initial setting up of the experiment with the graphical user interface (GUI) specifically developed and used within the research to aid in the exploration and analysis of the data and results. What follows is the descriptions of how the algorithm was implemented within MATLAB from the Map reduce stages to canopy clustering, and the final classification of the profile of the groups of loads.

8.2. MATLAB GUI Development

A custom MATLAB GUI was developed to aid in the exploration of the data. The GUI design enabled the different stages of the canopy clustering algorithm to be displayed, and the monitoring of the clustering process could be seen through each of the different stages of the algorithm. The GUI components were designed around the following criteria:

- Monitoring of the overall initial canopy clustering without the use of map reduce
- Display of the data points within each of the mappers and their relevant canopies
- Final canopy creation from the map reduce algorithm
- K-means clustering output and K-means centres from the final cluster allocation algorithm within the final canopies
- Selection of different tests that have been completed for analysis
- Variable methods of choosing the radius values of the canopies T_1 and T_2
- Selection of the harmonics that were being analysed within the canopy analysis

8.3. Canopy Clustering Overview

The flow chart shown in Figure 8-1 shows the overview of the canopy clustering algorithm. The following sections explain in greater depth how the algorithm is implemented.

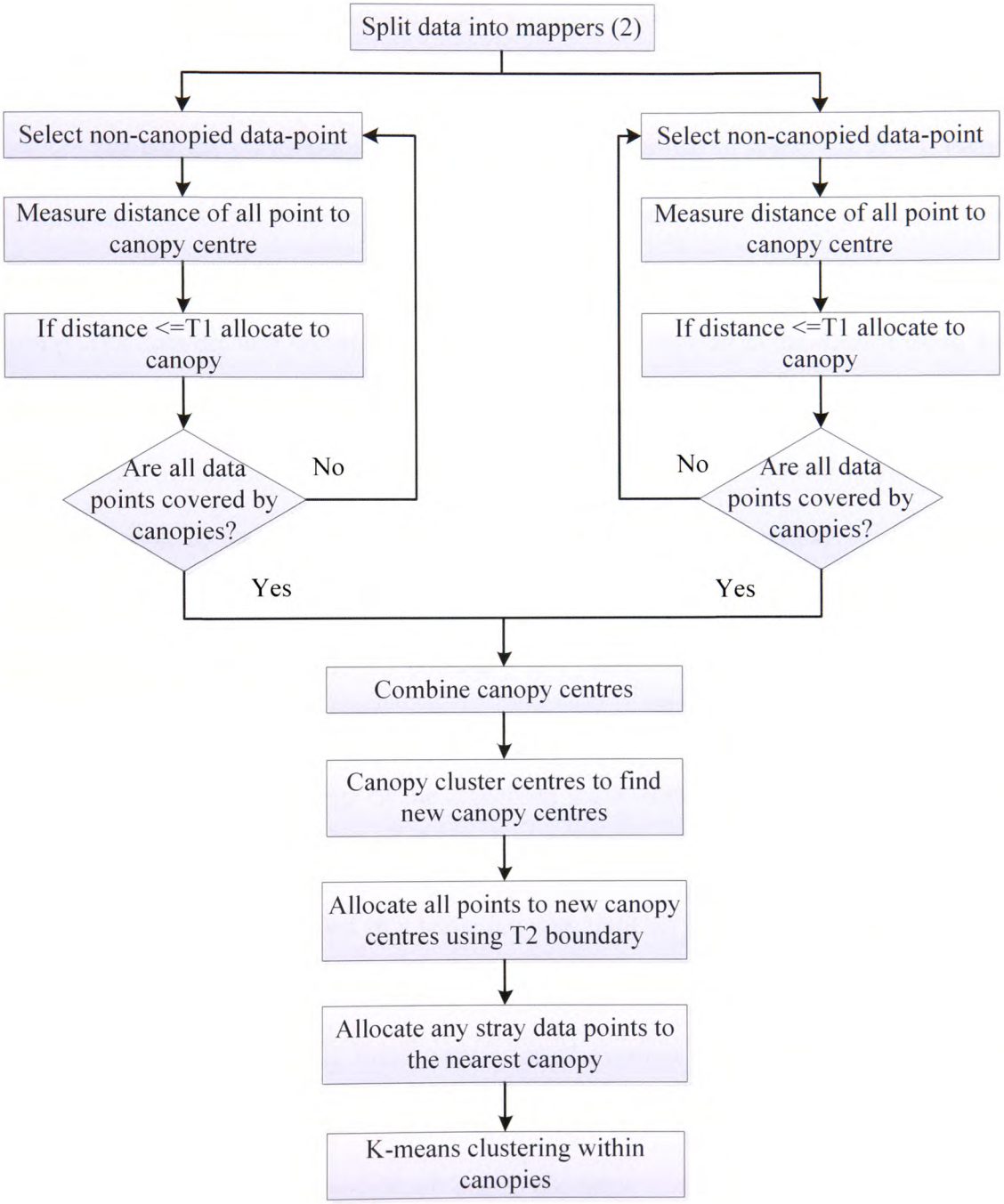

Figure 8-1 – Canopy Clustering Overview

8.4. Initial Canopy Clustering

The canopy clustering algorithm was used to separate the whole datasets in the first instance. This was initially completed to test the accuracy of canopy clustering within NILM before methods such as Map Reduce are used to aid in the improvement of the results. The initial canopy clustering used the whole dataset to determine the clusters and data point allocation.

The canopy clustering algorithm, which can be found in Appendix D – Canopy Clustering Algorithm, starts by using the values of T1 and T2 which were either fixed or determined as a ratio of the value of the canopy centre to the range of the expected group. The data set was sorted out by initially selecting a point in the dataset using the randperm function within MATLAB and using this as the centre of the first canopy. All other points within the dataset were then measured from this centre and if the distance from the point to the centre was less than T_1 then the point is said to be within that canopy, and could not be used as the centre point for further canopy creation. If the measured distance was within the boundary of T_2 then the point was said to belong to that cluster, but due to its distance from the centre it may be used as another canopy centre. The code for this stage of the algorithm implemented within MATLAB can be seen below in Code Sample 8-1.

This process continued in a recursive way until all of the points within the data set was covered, resulting in a large number of overlapping canopies. When this was considered on its own merit, there were a lot of overlaps with canopy centres within close proximity, and gaining a final result of cluster data that could be used to build up classification profiles became difficult due to the dense overlapping of canopies. To aid in the cleaning up of the overlapping canopies, the process of map reduce had been implemented to gain a better perspective of the data set.

```
function [center]=canopyCluster(t1,t2,x,y,z)
ref=1;
if(z==1)    %Use the range for the clustering
    centertemp=ones(size(x));
    for count=1:1:length(x)
        if (centertemp(count)==1)    %Can be used as canopy centre
            centertemp(count)=2;     %Used for Canopy Centre
            center(ref)=count;
            ref=ref+1;
            for count1=1:1:length(x)
                if(centertemp(count1)==1)
                    temp=sqrt((x(count)-x(count1)).^2+(y(count)-
y(count1)).^2);
                    if temp<=t1;  %Within the radius (half of the range)
                        centertemp(count1)=0;    %Cannot be used as canopy
center
                    end
                end
            end
        end
    end
end
```

Code Sample 8-1 – Canopy Creation

8.5. Map Reduce

The map reduce function was implemented to remove the redundant centres that were present within the results of the canopy clustering algorithm that was applied across the dataset as a whole. This implementation separated out the data set into two or more datasets of the same size, and the separation of the data points was completed using the MATLAB randperm function which ensured no bias had been introduced into the system. The code for the initial separation of the data is shown in Code Sample 8-2.

```
function [x1, x2,g1, y1, y2,g2]=mapreduce(x,y,g)
a=randperm(length(x));
count=1;
for i=1:2:length(a);
    x1(count)=x(a(i),1);
    y1(count)=y(a(i),1);
    g1(count)=g(a(i),1);
    count=count+1;
end
count=1;
for i=2:2:length(a);
    x2(count)=x(a(i),1);
    y2(count)=y(a(i),1);
    g2(count)=g(a(i),1);
    count=count+1;
end
```

Code Sample 8-2 – Map Reduce

This operation actually separated the dataset into the two individual map reducers, which consisted of the variables x1, x2 which were used for the x-axis, y1 and y2 used for the y-axis, and g1, g2, represented the actual groups of loads that the data point represent. The groups were included to enable a visual representation of the loads to be seen within the canopy clustering algorithm. The 'randperm' function was built into MATLAB and was used to generate an array of random numbers that are non-repeating of a specific length, and here the length of the variables was used to determine the size of the random number variable.

Each of the individual datasets created from the map reduce function, wre used to create canopy clusters in each of the sets. The map reduce function could be split into more than two datasets, with the amount of sets created depending upon the hardware that was used for the computation, as it was parallel within its operation. The present set up was completed with only two datasets as the sets are small enough to be completed on a desktop PC without any issues surrounding computation time, and was proposed to serve as an example of how the function would work.

To be able to remove the redundant centres, the new canopy centres were combined from each of the mappers. These canopies would cover all of the data points,

and there would be many more boundaries created covering the set, with many centres within any T_1 boundary. This redundancy of centres allowed a reduction of the canopy centres by conducting canopy clustering on the centres only, which would remove any canopy centres that had been created within the boundaries of other centres that were shorter in distance than T_1.

Once the new canopy centres had been found, the new canopies were created using the variables T_1 and T_2. The process of allocating the data points to the canopies is started using the canopy centres found from map reduce.

One underlying issue that surrounded this method of canopy clustering using map reduce, was that after the reduction of canopy centres, the whole dataset was not going to be covered by the canopies using the T_2 boundary. This was due to the centres moving from the original position. This problem was addressed by measuring the distance of the points outside the boundary to the centres of the canopies, and allocating them to the nearest boundary.

8.6. K-Means Clustering Within Canopies

8.6.1. K-Means Clustering Overview

The flowchart shown in Figure 8-1 shows the overview of the algorithm for finding the final clusters found using canopy clustering using the k-means approach.

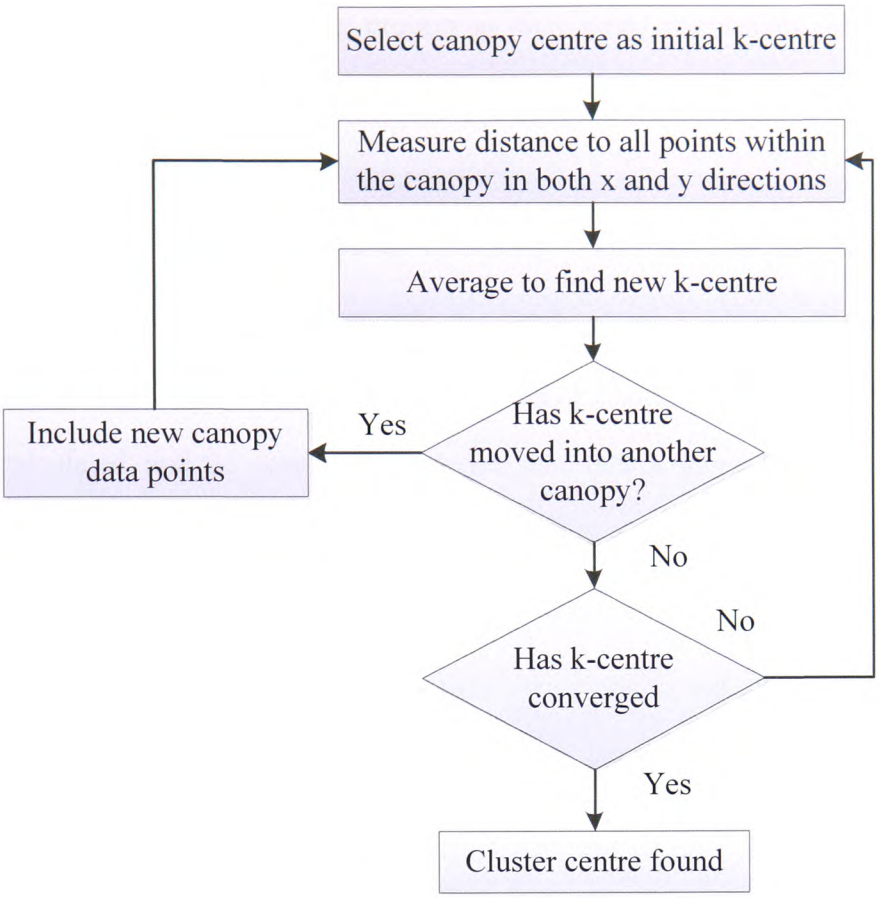

Figure 8-2 – K-Means Clustering Overview Flowchart

8.6.2. K-Means Algorithm Description

To be able to define the final clusters that were used for the building of the profiles that would describe the loads, the data points within the canopies would need to be collated to define the final clusters. Each of the canopies that had been developed to cover the dataset has had the variables T_1 and T_2 to be able to cover the final clusters, and this process has been described within 7.8.

The k-means clustering was where the real time-savings were made when using this approach to NILM. The clustering algorithm was carried out within the canopy alone, and since to the canopy clustering process separating out the data into distinctive subsets, there was no need to consider the data points that reside outside of the subset.

This reduced the clustering process as there were less distance calculations needed for each iteration due to its recursive nature; if this was carried out on a large dataset the processing time would become excessive.

The method of k-means clustering within the canopies was implemented with each of the canopies starting with its own k-centre which was allocated at the centre of the canopy. The calculations of the distance to all of the points within the canopy were calculated, and the new k-centre determined. The k-centre was free to move within the canopy, but the important feature to note was that there was the possibility that the k-centre would move into the vicinity of another canopy boundary. If this occurred, the k-centre could be said to belong to both the canopies, and as such the data points within the overlapping canopies needed to be taken into consideration when calculating the k-centre.

By allowing the k-centres to move freely across overlapping canopies, this alleviated the issue of more than one canopy covering a cluster that required grouping together, and effectively merges the canopies into one. The algorithm is completed when the k-centres converged within the canopy which, due to the fact that there was only one k-centre within each of the canopies, was efficient when compared to managing multiple k-centres. Only in the instance of the merging of two or more canopies does the process become data-intensive and the calculation time for the k-centres increased.

Each of the data points was finally allocated to a k-centre, and this was said to be the final clusters for the groups of loads that were found. Each of the groups were then analysed to build up a profile that would be used for the classification of new points that were taken when the system was in full operation. These profiles include the ranges of the groups within the different harmonic domains, cluster centres and

distribution of the clusters. All these variables made up the classification criteria for the groups of loads that were being analysed.

8.7. Conclusions

This chapter has detailed the processes involved within canopy clustering. The algorithms for canopy clustering have been developed within MATLAB, which was ideal due to the ability to develop custom user interfaces, which had been integral to the exploration of investigating the canopy clustering algorithm. By using a GUI, a quick overview of the algorithm could be seen and aided in the constant changes that were applied to the algorithm.

The use of canopy clustering on the dataset as a whole has been described, which was the logical approach to take at the beginning of the research. When further investigating the approaches of applying the algorithm to a large dataset, there were many redundant, overlapping canopies that overcomplicated the problem during the process of identifying the final clusters.

To alleviate the issues surrounding the redundant canopies, Map Reduce was used on the dataset. Map reduce was the process of splitting the dataset up into smaller subsets to allow canopy clustering to be completed on these smaller subsets. The implementation of the map reduce methods within canopy clustering enabled the problem to be solved in a parallel fashion, and the number of mappers that were chosen was dependant up on the number of computer processor cores available. In this instance the amount of mappers chosen was two, and were computed within MATLAB in a serial fashion, as the size of the datasets was small enough to be computed on a desktop machine.

The final output of the mappers was then used to compute the final canopies of the whole dataset. This was completed using canopy clustering on the canopy centres obtained from the mappers, and made up the reducer part of the map-reduce algorithm.

The final canopy centres then fitted the clusters that were present within the dataset, without the large amounts of redundant overlapping canopies. Due to the change of the canopy centres there were a number of data points within the data set that were not covered by the canopies, and these could be allocated to the nearest canopy for further computation.

The method of integrating different clustering technologies such as canopy clustering with traditional methods of clustering like k-means, allowed very large datasets to be analysed and classified in a streamlined manner, and as such, lend themselves to be used within the field of NILM.

Chapter 9. Load Classification

9.1. Introduction

The load classification chapter looks at how the groups of loads can be broken down into component parts using the information obtained from the load operating characteristics and the Fourier analysis of the total load current. The initial section looks at how trigonometric identities can be used within the classification process. The outputs of these functions were used in conjunction with a rules based system that had been developed to enable the classification of individual loads within a group.

9.2. Load Combinations

When considering the frequency spectrum which was used for analysing the total load currents, the frequency components of interest were all of multiples of the fundamental frequency of 50Hz. These frequencies were present within each of the various loads, with varying amplitudes, and phase's. The theory of linear combinations could be used for analysing the makeup of the total frequencies within the system.

When adding sine waves of the same frequency and different phase, the resultant signal will be a sine wave of the same frequency and a different phase. The way in which these sine waves interact was important from the stand point, that simple addition of the amplitude of the signals cannot be used independently of the phase of the signals, and their interaction is more complex. The addition of two sine waves of the same frequency can be calculated using (9-1).

$$a\sin x + b\sin(x+\alpha) = c\sin(x+\beta)$$

$$c = \sqrt{a^2 + b^2 + 2ab\cos\alpha} \qquad (9\text{-}1)$$

$$\beta = \tan^{-1}\left(\frac{b\sin\alpha}{a + b\cos\alpha}\right)$$

Using this representation, the frequency components of the individual known loads could be combined to equate to the frequencies shown within the groups of loads that were found within the total load current. Conducting load disaggregation within the groups of loads, instead of the whole dataset would require less computation overall and provide a streamlined process of load classification.

To be able to correctly identify the loads within the group, the two-dimensional domains that were used within the canopy clustering to obtain the groups required further investigation. Each of the groups that had been identified would need to be defined as a group of loads and therefore each of the pairs of frequencies used to make up the two-dimensional space would need to be analysed in tandem to identify the individual loads that make up the group.

The classification process becomes computationally expensive, due to the amounts of loads that need to be taken into consideration, and how loads interact. To be able to identify the loads within a group, the summation of all the frequency components would need to equate the final frequency components of the total load current. By using iterative procedures, each of the frequency components could be calculated from the known loads, and the final summation of the individual loads would fall into the boundaries of the group within the specific frequency domain.

Initially the fundamental frequency and third harmonic was considered, and iteratively each of the signals were added together to find the combination of loads that fits the profile of the boundaries within the fundamental domain only. It should be noted at this stage that there may be multiple groups of loads that would fall within the specific group, but the defining characteristics would be when the loads were analysed at different frequencies, and the groups would diverge into different boundaries. As an example two loads may have the same current amplitude within the fundamental frequency, but when considering the third harmonic, the amplitudes could be different.

Therefore the ability to analyse the amplitudes present within the higher harmonics allows for separation of loads that show to be similar within the fundamental frequency.

Evaluating the characteristics of the loads should be completed within the complex plane, as the addition of complex numbers is a simple process, and just requires the amplitude and phase information of the total load current and the currents of the individual loads to be converted to the complex domain. The complex numbers are then summated to obtain the final coordinates, which can then be converted back to polar form. The calculations for the conversions can be seen in (9-2).

$$\begin{aligned} real &= r\cos\beta \\ imaginary &= r\sin\beta \\ r &= \sqrt{real^2 + imaginary^2} \\ \beta &= \tan^{-1}\frac{imaginary}{real} \end{aligned} \tag{9-2}$$

Once the amplitude information of the groups was obtained for the two frequencies, the groups that had been defined within the canopy clustering could then be compared to the amplitudes to ascertain which of the group's boundaries that the totals fall into. If there were instances where the addition of the combination of the loads did not fall into any of the group boundaries, it was said that the specific combination of loads was not present within the dataset, and could be ignored for further analysis.

The process was completed for the other two-dimensional spaces that had been used within the canopy clustering process. The combinations of the loads that had been determined as not present within the first stages of the calculation were not taken into consideration from this point onwards, thus speeding up the calculations that need to be performed.

Once all of the groups within the different domains had been allocated to different combinations of loads, a final profile could be created for the loads. The

profile of each of the loads would be determined by its group membership within each of the different domains, and therefore the process of deciding if a load is present at any particular point in time would be reliant upon the present group membership of the harmonic content for that sample period.

The load classification algorithm would only need to be performed after the initial training period and again when new loads were added to the system, and therefore the computationally expensive parts of the system were only required to be conducted rarely. Even though there were a lot of calculations to be completed, this could be done in parallel to the data acquisition to allow continuous monitoring of the load activity within the premises.

By finding out the individual loads that were within each of the groups, the ability to measure the energy usage of the load was viable. It would be known how each of the individual loads performed within each of the groups, so the energy usage of the load could be calculated. This energy usage for the loads was then used for monitoring purposes by the end user, which could be used for displaying when and where energy is being used.

9.3. Conclusions

One important concept discussed within this chapter was the interaction of the different loads and the way the phase changes of the harmonics alter the overall phase of the total load current. By observing the different phase changes of the individual loads, the identity of the loads within the groups defined within canopy clustering could be computed. It should also be noted that the phase response of the digital filter did not need to be taken into consideration, due to the zero phase shift function within MATLAB.

The use of displaying the frequency components of the signals within the complex plane converts the problem into a case of vector addition in the real and

imaginary components, which simplifies the process. The conversion back to polar form allows a comparison between the additions of the individual loads to be made against the pre-existing groups. Once the loads have been identified, the power of the individual loads can be calculated and stored to aid in feedback to the end user as to when and where energy savings can be made.

Chapter 10. Results

10.1. Introduction

This chapter details the results from the research. Initially the methods were applied to purely resistive loads, and shows how the canopy clustering algorithm could be used on the data. This was later applied to non-linear loads where the information in the subsequent harmonics played a more important part in the clustering. The chapter further explains how the canopy creation could greatly affect the final results.

10.2. Resistive Loads

Resistive loads were notably different from non-linear loads in that all of the energy being consumed by the appliance was contained within the fundamental frequency. Due to this the testing of the devices that were resistive only were only considered within the fundamental plane. The initial processes of the algorithm was used to determine the operating conditions of the loads, and for the first example the use of two common household items were tested, a toaster and kettle.

Both of these appliances were resistive, and operated by converting the energy consumed into heat through resistance. The two loads were connected to the DAQ device and used, with the current and voltage waveforms recorded. These were then imported into MATLAB where they could be further analysed. Each of the signals recorded was put through the zero phase filters to remove any unwanted high frequency noise. The signals were then processed through Fourier analysis to convert them into the frequency domain.

The output of the Fourier analysis provided the harmonic content for each of the sample window sections, which covered a one second period. The data was plotted in a time varying three-dimensional plot for both of the load currents and the total current.

Figure 10-1 shows the frequency output of the FFT within MATLAB. It can be seen that the load was purely resistive, as all of the current is drawn in the fundamental frequency with a small residual current being displayed in the 250Hz harmonic. The actual current draw of the toaster varied over time, and this variation will be taken into consideration when creating the profiles for the toaster.

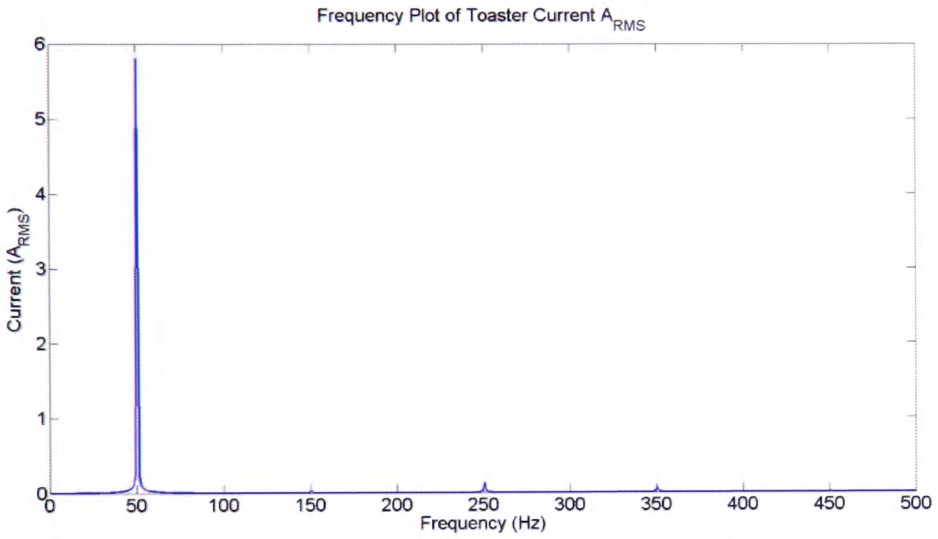

Figure 10-1 – Frequency Plot of Toaster Current Draw

When applying the same methods to the kettle, a similar plot could be found, and this was due to the resistive nature of the device. These two devices could be seen to have a simple operation and only work in an on/off state and therefore the profiles for resistive loads only need to consider the on-state. Figure 10-2 shows the frequency content of the kettle. It could be seen that the operating conditions were very similar, but there were some small harmonic components that were present, however these components were insignificant when compared to the actual current consumed within the fundamental frequency. Due to the small size of the higher harmonics, they are not required for the creation of the profile of the kettle and only the fundamental frequency is used.

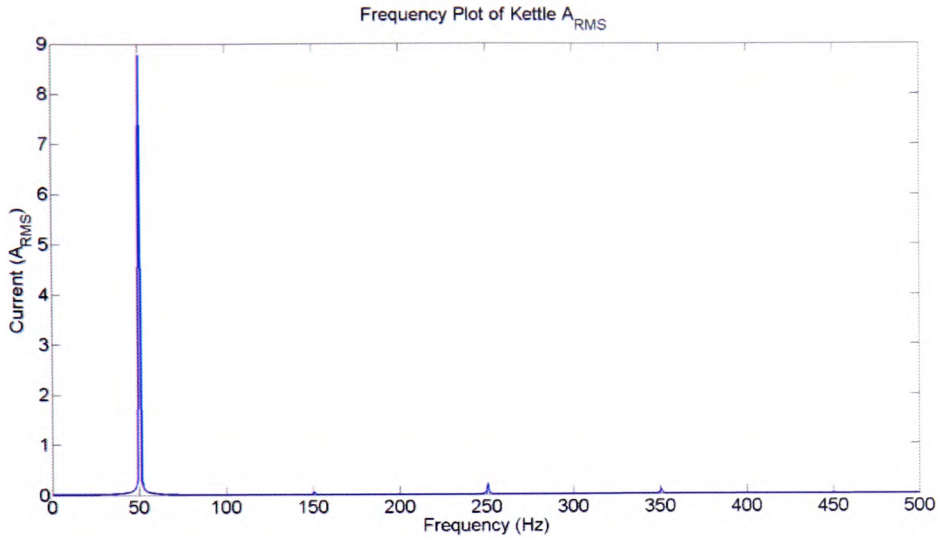

Figure 10-2 – Frequency Plot of Kettle Current Draw

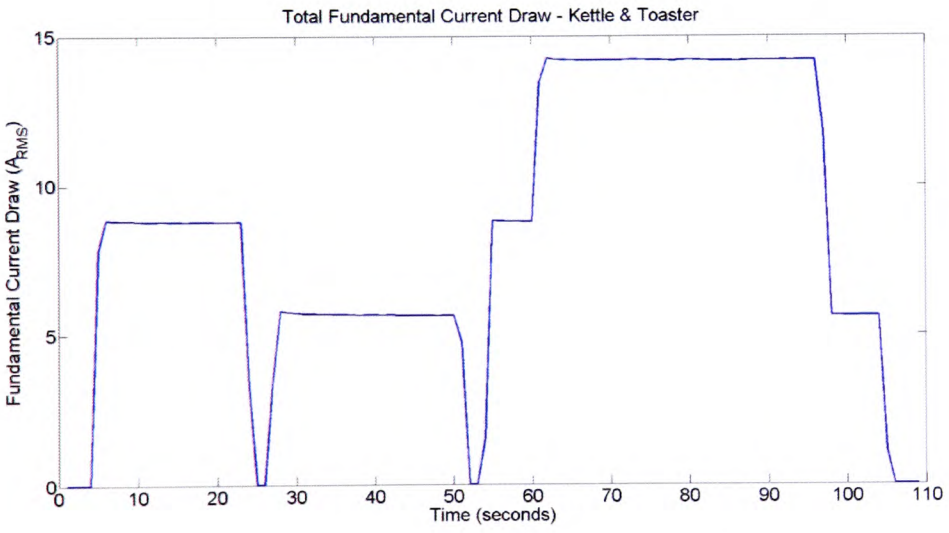

Figure 10-3 – Total Load Frequency Plot over Time (seconds)

The total load current had been recorded, and passed through the same procedures to provide a frequency plot over time. Figure 10-3 shows the total load current over time. From this plot, the different loads that were on their own can be seen with the kettle being turned on and off at the beginning of the plot, the toaster was then turned on and off, then both loads are used at the same time. The sections in between the peaks within the plot were attributed to the transients of the loads being turned on or

off mid-way through the sample period, and therefore they are neither at zero current or maximum current. When considering the higher harmonics, as expected there are none present of any significance due to the resistive nature of the loads, and therefore only the fundamental frequency was taken into consideration for classification.

To be able to correctly identify the individual loads from the total load current, each of the individual loads needed to be marked up for their operating conditions, and within this test, there were only two conditions which need to be identified, the loads are said to be either on or off. Both loads were calculated and then the final total load group was also identified which was a combination of the two individual load groups.

This was completed by allocating each of the loads a two digit identifier for each of the loads operating conditions and was then combined to create a four digit code for the group. The first two digits symbolising the first load, and the latter two digits represent the second load. As stated previously there were some points that were classed as transitional points, and those were labelled with a zero and are not used for the classification process, but included into the final usage reports when the loads have been classified.

Toaster	Kettle	Group Allocation
Off	Off	1111
On	Off	2111
Off	On	1121
On	On	2121

Table 10-1 – Load Allocation Codes

The load groups that have been allocated can be seen Table 10-1 and have been obtained when analysing the loads individually. This ability to analyse the loads on an individual basis enabled the final results of the canopy clustering to be checked and enabled better visualisation of the data when plotted in a two dimensional plane, for use in canopy clustering. To better understand how the codes could be applied to a graph of

the data in a two dimensional space, the codes shown in Table 10-1 are used to display the groups within the scatter plot of the fundamental and third harmonic and can be seen in Figure 10-4. The third harmonic content seen within the plot is insignificant when compared to the amplitude of the fundamental frequency currents and therefore can be seen as a resistive load.

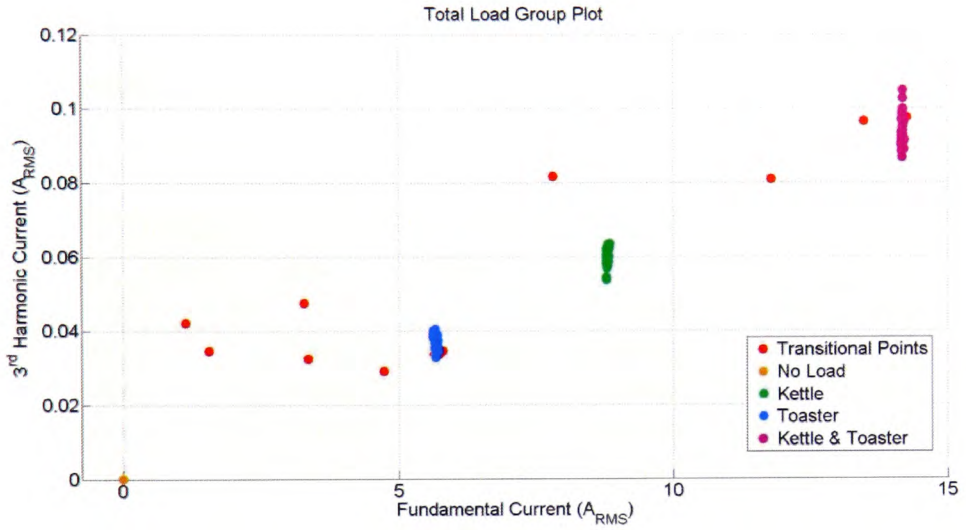

Figure 10-4 – Scatter Plot of Total Load Current with Load Codes

Here the way in which the loads were interacting can be seen, the different operating conditions were described by the codes indicated in Table 10-1, and the transitional points were represented by zero (red point). By colour coding the plot with the use of the codes, it could be seen that the groups have been identified by the algorithm, with the cyan point (group 1111) displaying the points in time when both loads are off. The kettle and toaster can be seen by the blue (2111) and green (1121) respectively. The combined loads are represented by the purple dots (2121).

By visualising the data in this way, it allowed for the next stages of canopy clustering. To be able to correctly use canopy clustering the relationship of the groups that were identified need to be considered. This was to ensure that the canopies are set

up correctly in a way that all of the intended data that was to be clustered, could be covered by a single canopy, even though there may actually be more than one canopy covering the desired cluster.

To better understand the relationships of the clusters, the data has been analysed to show the make-up of the cluster, which included range, mid-point and average within the harmonic domains. With this load being resistive, only the fundamental components are considered and can be seen in Table 10-2 which are the results of the tests conducted over 109 seconds.

Group	Count (sec's)	Min	Max	Average	Range	Ratio
0	21	0	50	5.119349	-50	-4.88343
1111	7	0.001285	0.001214	0.00124	7.07E-05	0.028504
1121	22	8.845546	8.779393	8.800868	0.066153	0.003758
2111	25	5.739902	5.613351	5.668175	0.12655	0.011163
2121	34	14.22581	14.16593	14.1901	0.059878	0.00211

Table 10-2 – Group Analysis Information

The table shows the values that have been calculated for the groups. The count variable represented the number of data points within the group, which was also an indication as to the period of time that the load was in operation. The min and max variables were the minimum and maximum current values within the groups, and the range was the difference between the minimum and maximum values. The average value was calculated as the summation of all the data points within the groups divided by the amount of points, and the ratio was the calculation of the ratio of the average value of the group to half the range (radius) of the group.

When considering the information presented for the expected clusters, a relationship between the centre points of the cluster and the boundaries can be found. These boundaries were important when it came to describing the canopies for the clustering algorithm as the distance used for the boundaries would greatly affect the

overall results obtained from clustering, and therefore the canopy boundaries had to be selected that represented the clusters contained within the dataset. To be able to find the relationship between the centre and the ratio, the basic fitting tool within MATLAB had been used and a plot of the data can be seen in Figure 10-5, where the average of the group within the fundamental domain has been plotted against the ratio of the average to the range. This relationship allowed for the calculation of the boundaries used to describe the canopies which were representative of the clusters found through the analysis of the individual loads.

Fitting the function of the relationship, the boundaries of T_1 can be found, with the relationship shown in equation (10-1). The boundary of T_2 can be described as a method of allowing for some variation within the canopies. When considering Figure 10-5, the fit of the function to the data points is good, and therefore the leeway needed to be sure that the whole of the cluster was covered by the canopy was covered by the function for T_1. The number selected for the value of T_2 would be selected depending upon the differences between the function and the measured values.

$$Ratio = 0.00016x^2 - 0.0041x + 0.029 \qquad (10\text{-}1)$$

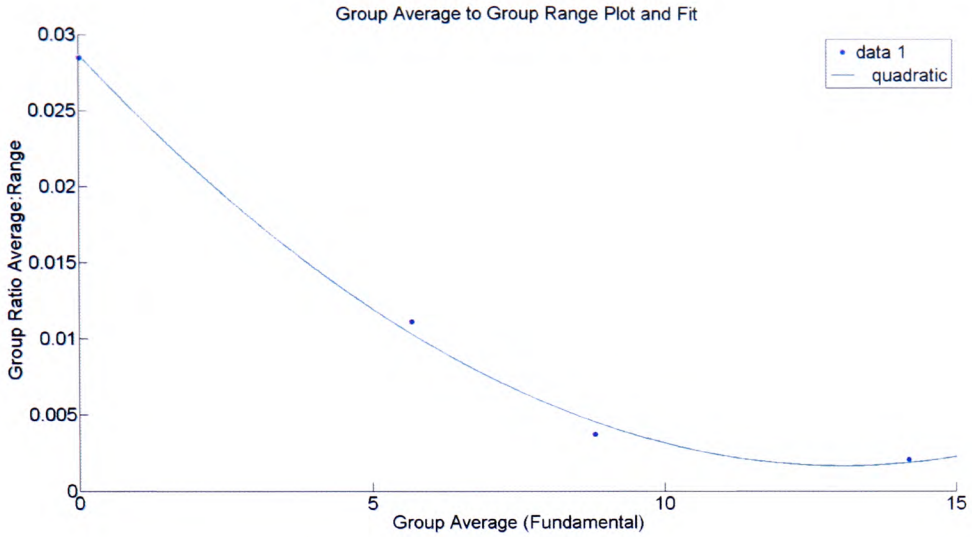

Figure 10-5 – Group Average to Range Ratio Plot

The relationship had been found that identified the size of the canopy boundary T_1 to the selected point that was being used for the canopy centre. This information was used when creating canopies within the individual mappers of the canopy algorithm. The mappers split the data set into smaller subsets that could then be processed individually and recombined to create an overall better fit of the data. This process could be seen by using the current data of the kettle and toaster as an example.

The data obtained was split using the inbuilt MATLAB randperm number generator. The data was then processed using the canopy clustering algorithm to find the canopy centres of the datasets sent to each of the mappers. At that point only the values of T_1 were considered due to the close fit of the function to the measured values. The first of the mappers was processed and the canopy centres were found, recorded with the T_1 boundary drawn and can be seen in Figure 10-6:

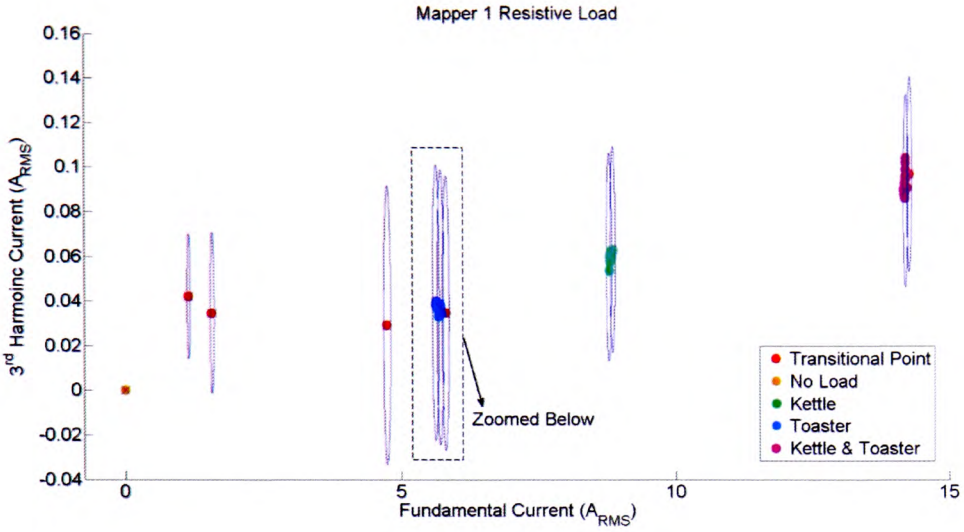

Figure 10-6 – Mapper 1 of Resistive Load (kettle and Toaster)

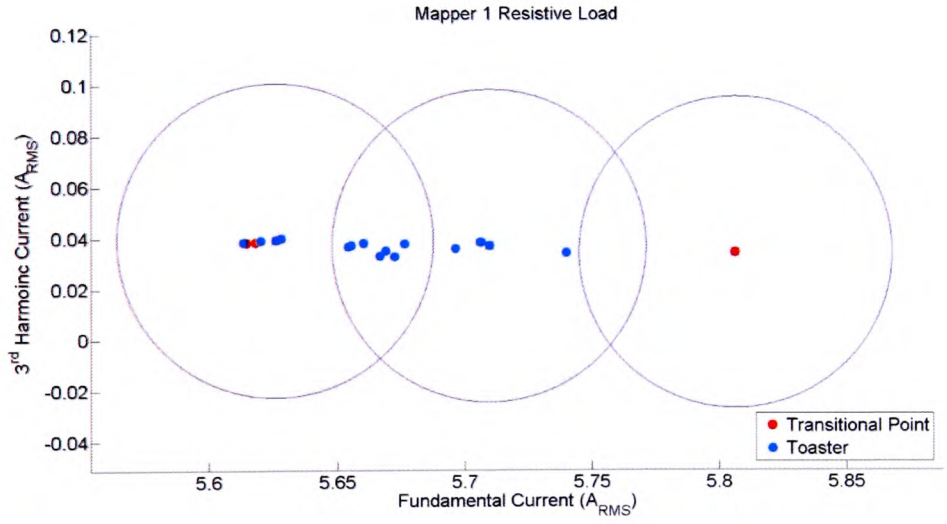

Figure 10-7 – Mapper 1 Zoomed Resistive load of toaster canopy

The canopies could be seen to have varying dimensions depending upon the canopy centre used. When zooming into viewing the individual canopies that were covering the different combination of loads, each of the canopies could be seen to be covered by at least one canopy. If more than one canopy was covering the data, there was sufficient overlap to ensure that the final canopies would become more representative of the data and can be seen in Figure 10-7.

The mapper had been used for the second half of the data, together with the plot shown in Figure 10-8, and exhibited the same characteristics as mapper one. The canopy boundaries could be seen to vary again with the function that had been used to describe the relationship shown in equation (10-1).

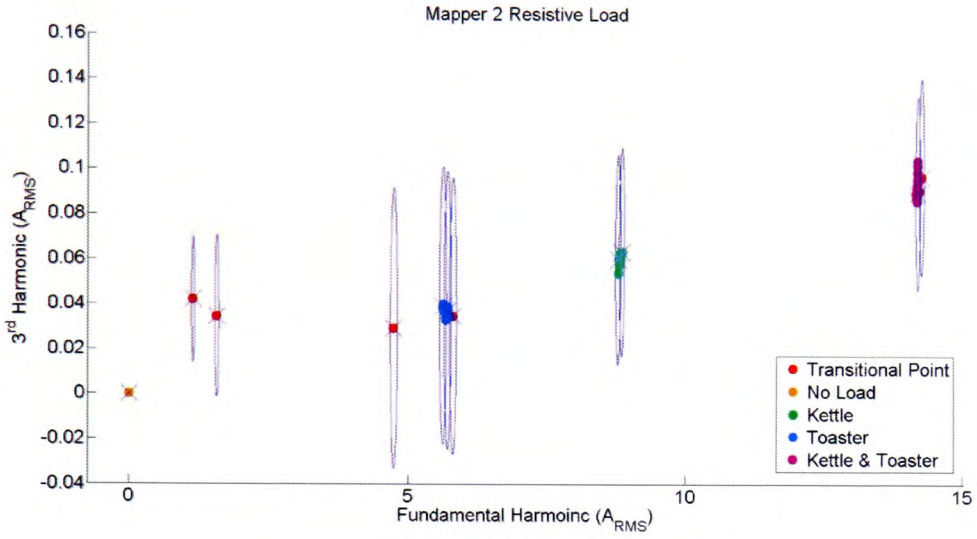

Figure 10-8 – Mapper 2 Resistive Load – Toaster and Kettle

The two mappers had been used to find the canopy centres and had shown to be able to cover all of the data points within the canopies. The information from each of the individual plots could be used to find the final canopy centres that were used for the final canopy clustering of the data to identify the groups within the dataset. This was completed by combining the canopy centres, and running the canopy algorithm on the actual canopy centres found from the mapper process. This process reduced the total number of centres used for the canopy clustering, whilst giving an overall better fit for the data.

The map reduce function within the canopy algorithm reduced the overall canopies that were required to cover the dataset by canopy clustering the centres found from the mappers, and averaging the centres to find the midpoint in two-dimensional

space for the new centre. The process had reduced the amount of centres used from the two mappers from 30 to 17, therefore close to a 50% reduction from the mappers. The final plot of the canopy centres and the T1 boundaries can be seen in Figure 10-9.

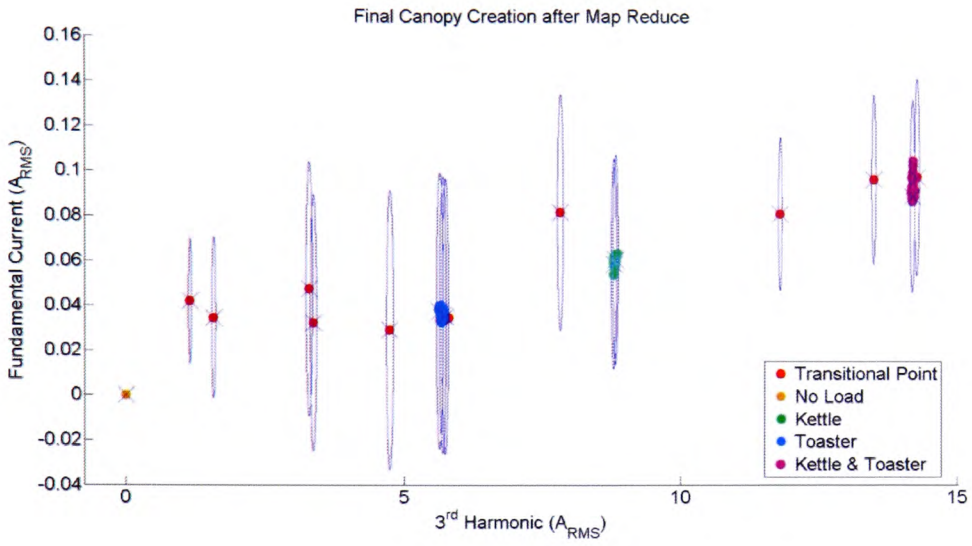

Figure 10-9 – Map Reduce Final Canopies

Many of the final canopies were attributed to the fact that there are many transitional points within the dataset, which were the points at which loads have been turned on or off in the middle of the sampling period. These points were no longer required for further analysis, and were no longer used within the canopy analysis algorithm. These points were easily identifiable by the fact that the canopy covering the transitional points only contained one data point.

During the map reduce phase, there were many points which were no longer covered by the canopies. This was due to the fact that the new canopy centres had moved, and the boundary was no longer big enough to cover the existing points. To alleviate this issue, the points that were unattached to canopies, were allocated to the nearest one. This ensured that all points were allocated to the canopies, and the obvious

transitional pointes were already covered by canopies, meaning at the stray points had a higher chance of being a member of the nearest canopy.

Each of the canopies were converted into their own variables, and within these variables the final stage of the algorithm was completed. To finally convey the profiles created by the algorithm, each of the points needed to be allocated to a final cluster. This was completed using k-means clustering within the canopies alone.

The actual canopies within the plot could be seen to be overlapping, and this overlapping came naturally with the process, as there was a random element to selecting the initial centre points. To ensure that the correct result was selected the overlapping canopies were combined to provide a wide canopy that covered the load that required classification. The combinations of the canopies that can be merged are shown in Table 10-3.

Canopy 1	Canopy 2	Canopy 3
1	2	11
3	5	0
4	9	0
5	0	0
6	16	0
7	0	0
8	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
17	0	0

Table 10-3 – Canopy Overlap

By combining the overlapping clusters and computing the final cluster contained within, the redundant clusters were removed, which essentially represent the same load within the harmonic domains. After the completing of the k-means clustering within the

overlapping canopies on the original dataset, the final load allocation can be seen in Figure 10-10.

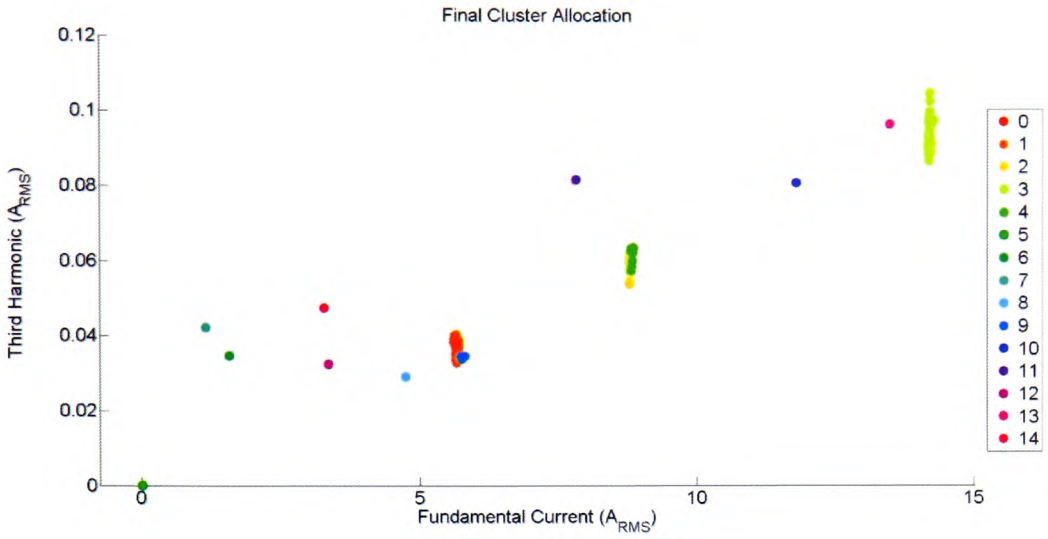

Figure 10-10 – Final Cluster Allocation

The final canopy allocation could be seen to be representative of the initial plot where the known loads were found. The four main groups of interest had been identified and to check the accuracy of the allocation of the data to the correct clusters, the original group allocation was checked. The four groups identified were when all loads are off, the kettle and toaster are on individually, and when both loads are on simultaneously. The results are shown in Table 10-4 and the accuracy of the results is good when considering that both the loads were in the on-state, and the individual loads are found with a good accuracy. The lower accuracy of the off-state of the loads was found to have a good accuracy, but not important, as the system will not be monitoring the off-states.

Initial Group	Member Count	Allocated Group	Member Count	Accuracy
1111	7	5	6	85.71%
1121	22	2&4	22	100.00%
2111	25	1	24	96.00%
2121	34	3	34	100.00%

Table 10-4 – Group Comparison Table

The data presented within Table 10-4 shows the comparison between the initial groups and the final group allocation. One of the important notes to make is that one of the groups that represent the kettle had been split into two adjoining groups within the final analysis. This has occurred due to the initial canopy clustering of the data. One of the canopies contained a transitional point and has therefore split the cluster into two adjoining groups.

The other factors to consider when looking at the clustering methods and its accuracy is that there are transitional points contained within the final clusters and this was due to them falling within the canopy of the final clustering allocation. It was impossible to remove these, due to the fact that they are within the canopy boundary and was seen as one group. These transitional points represent the start up or turn off of the load and therefore did not have any significant importance, as these could be later confirmed by analysing the points that follow the transitional point and then allocating the energy usage to the load that was found.

10.3. Non-Linear Components

The previous section analysed the canopy clustering algorithm using the test data to create the relationships for the canopies, which was the fundamental basis of the research, and an important requirement of the overall process. To further test the system different loads were used and processed through the algorithm. The process looked at not only resistive loads but brought other loads into the system which had significant power draw within the harmonic frequencies.

10.3.1. Test with Non-Linear Loads

For the second test three different loads were used, which included other traditional common household objects such as a microwave and an iron. The microwave can be classed as a non-linear load, and the iron is said to be linear. The individual load current frequency plots are shown in Figure 10-11, the microwave and Figure 10-12 the Iron.

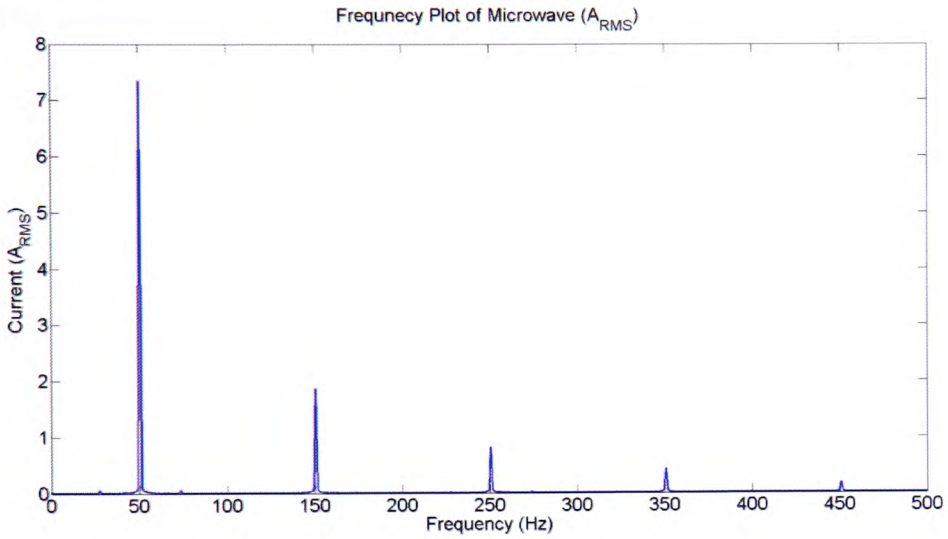

Figure 10-11 – Microwave Current Draw Frequency Plot

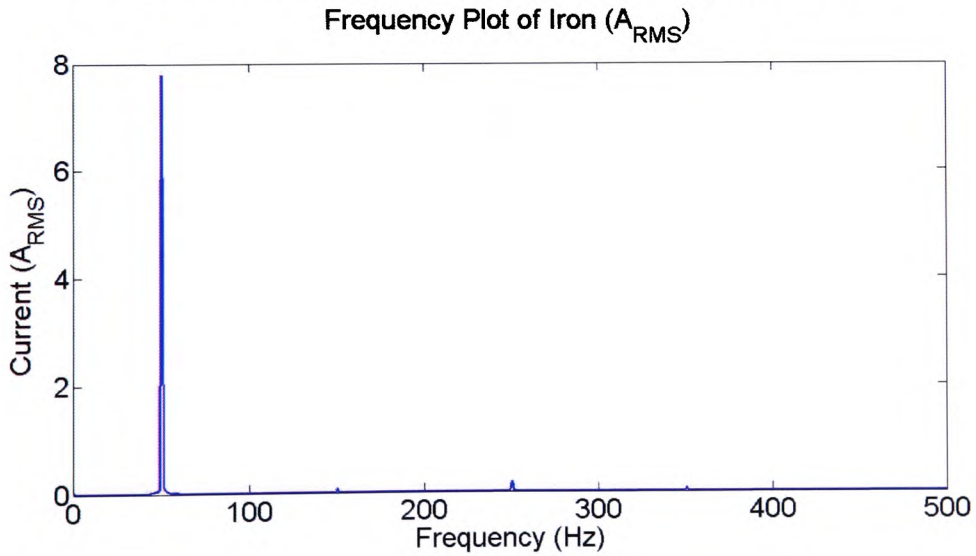

Figure 10-12 – Iron Current Draw Frequency Plot

The difference between the linear and non-linear loads can be seen when comparing the harmonics in the frequency plots of the microwave and the Iron. The frequency plot of the microwave showed that the current is present in the fundamental

and third, fifth, and seventh harmonic at amplitudes that are noteworthy, and to a lesser degree the ninth harmonic. When compared to the frequency plot of the iron, all of the current draw can be seen in the fundamental frequency. The difference in the frequency plots showed that the microwave has a greater number of features that can be used for classification compared to the non-linear load shown by the iron.

The frequency plot of the microwave shows that the current is contained within the odd harmonics only, and this is attributed to the fact that if the final load consumes a symmetrical current in the positive and negative in the time domain, then only odd harmonics will be present in the frequency domain [108].

The sampling period of the microwave and the iron had been conducted so that each of the loads can be seen individually and combined. The difference within the frequency spectrum can be clearly seen within Figure 10-13. When the microwave was on either alone or when the iron was on, the higher order frequency could be seen within the third harmonic, whereas the iron alone only consumes energy within the fundamental component of the spectrum. It should be noted that at the beginning of the sampling period neither of the loads were plugged in, therefore zero current was drawn, but when the microwave was plugged in, there was some residual current being consumed. This residual current is due to the microwave being of the digital type and consumed some current for the LCD display.

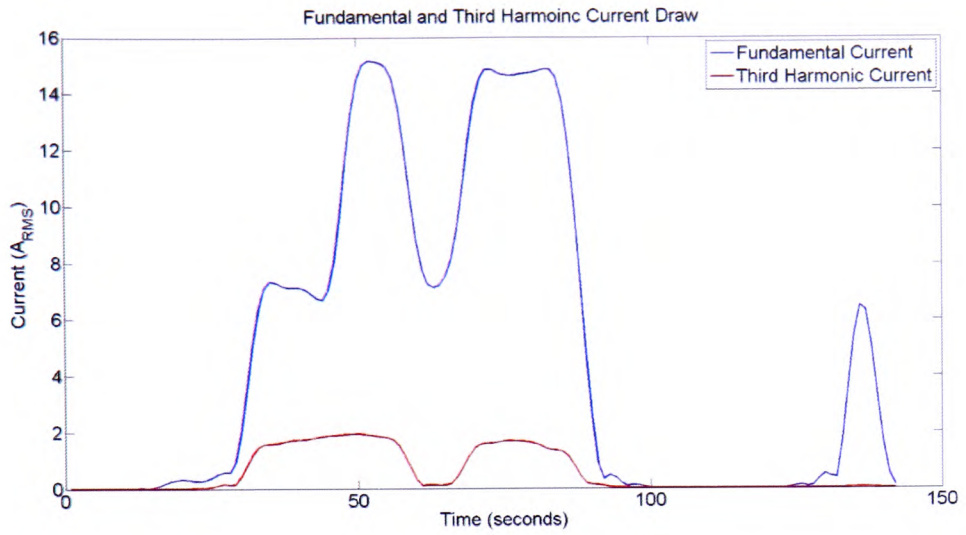

Figure 10-13 – Fundamental and Third Harmonic Current Total over Time of Microwave and Iron

One of the main issues that were countered with the use of non-linear loads was that there were large components within the higher harmonics. The plot for the fundamental and third harmonic can be seen to have different characteristics to that of the resistive loads as seen in Figure 10-14.

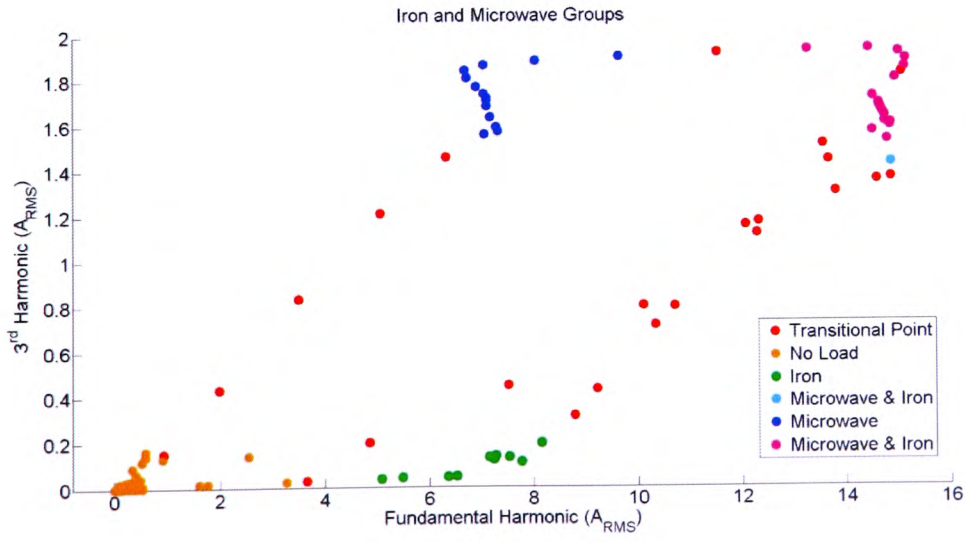

Figure 10-14 – Iron and Microwave Grouping

This plot clearly identifies the issues surrounding the clustering of non-linear loads with the way the data is spread, and does not display any clearly defining boundaries. The reason for this was the operating conditions of the microwave. When considering the microwave, there are many different power settings that can be used, and in this instance the high power and defrost mode were used. The operating codes for the different loads and there conditions can be seen in Table 10-5.

Microwave	Iron	Code
Off	Off	1111
Off	On	1121
On	On	2221
On	Off	2311
On	On	2321

Table 10-5 – Microwave and Iron Operating Conditions

The different operating conditions of the microwave were such that when it was on high power, it was continually drawing power, but when on defrost mode, the power was actually cycled on and off. The on power state for the microwave while in defrost mode drew the same power as the high power mode. This means that there are many transitional points from on to the off state during the defrost mode, which could be seen in the scatter diagram of the current draw.

When considering the iron, there were two operating conditions, which were described as on or off. The iron heats up to its operating temperature, as defined by the dials on the appliance. When the temperature was reached, it was turned off, and no longer drew any power, until the temperature dropped below a certain threshold, and then it reheated the plate. This was another example of a switching appliance, which made the final clustering scatter plots crowded with random points due to the amount of transitions of the appliance.

When considering the difference seen between the purely resistive and the non-linear loads, the spread of the current draw was shown to be different within Figure

10-4 and Figure 10-14 and therefore a different relationship for the canopy boundary needed to be found. The non-linear plots showed that the spread of the different groups is similar regardless of the current draw, and therefore the relationships described for the resistive loads no longer held true. This meant that there was a requirement for there to be another method of creating canopies for non-linear loads, which was not tied to the average current of the groups.

When considering the relationships between the groups, and the canopy boundaries that needed to be described, the range of the group had been considered as the function of the average. This allowed for the boundary to be mapped to the centre of the groups, and was considered a better fit than the ratio, which was used within the resistive load due to the larger spread of the data within the groups. When removing all of the points that were classed as transitional points, and mapping the average values of the groups to the range of the groups defined, then a relationship could be found shown in equation (10-2) and Figure 10-15.

$$Range = -0.0077x^2 + 0.019x + 3.3 \qquad (10\text{-}2)$$

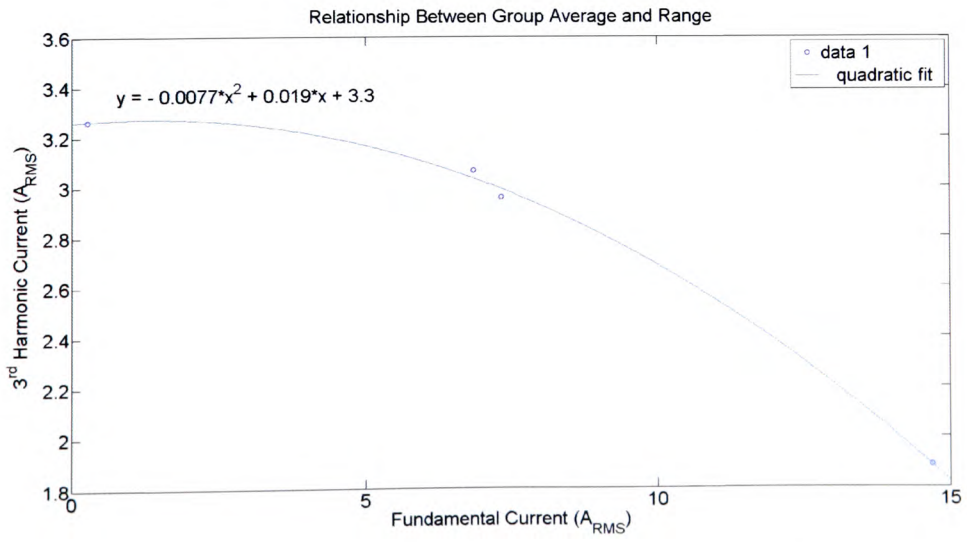

Figure 10-15 – Non-Linear Range to Average Relationship

The relationship had been found by using the basic fitting tool within MATLAB, and for the data provided a good fit, with the transitional points being removed. The relationship was used within the canopy clustering algorithm for the distance T_1. Equation (10-2) shows the relationship for the average to the range, and therefore the range needed to be divided by two to obtain the value for the radius boundary T_1.

By applying the T_1 boundary to the data the initial canopy clustering algorithm can be seen in Figure 10-16. From the plot it can be seen that the groups of interest, ignoring the transitional points, were covered by the canopies. There were a small number of data points that are outside of the canopies that contain the main group, but when compared to the amount of data points within the canopy would only provide a small reduction in accuracy. The transitional points were covered by most of the canopies, and would end up being grouped with the final clusters due to the amount of transitional points that occurred due to the nature of the loads. The relationship between the average value of the clusters and the radius boundary T_1 can be seen clearly within the figure, and the varying canopy size provided a better fit for the data overall.

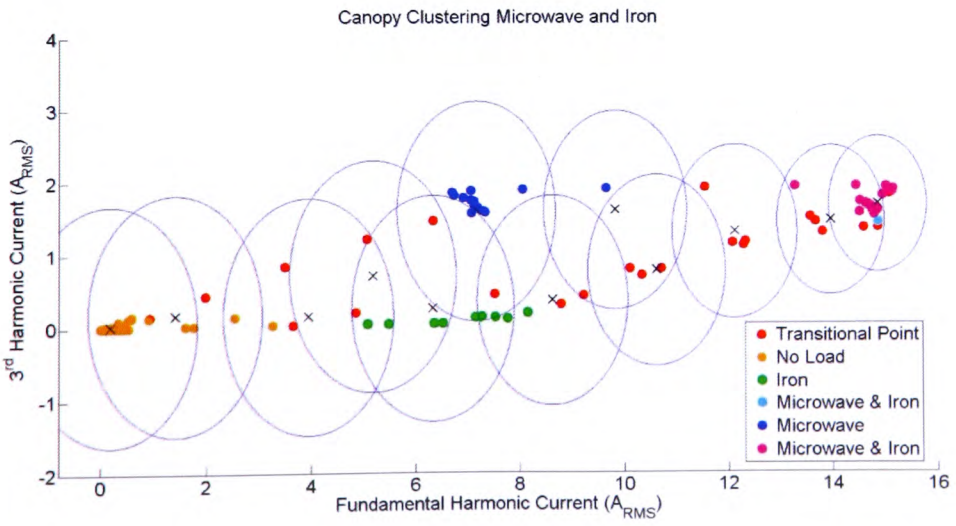

Figure 10-16 – Initial Canopy Clustering Microwave and Iron

To finalise the actual clustering of the loads, the data was passed through to the k-means section of the algorithm. This combined the overlapping canopies, and provide the final clusters that were used for the profiling of the loads.

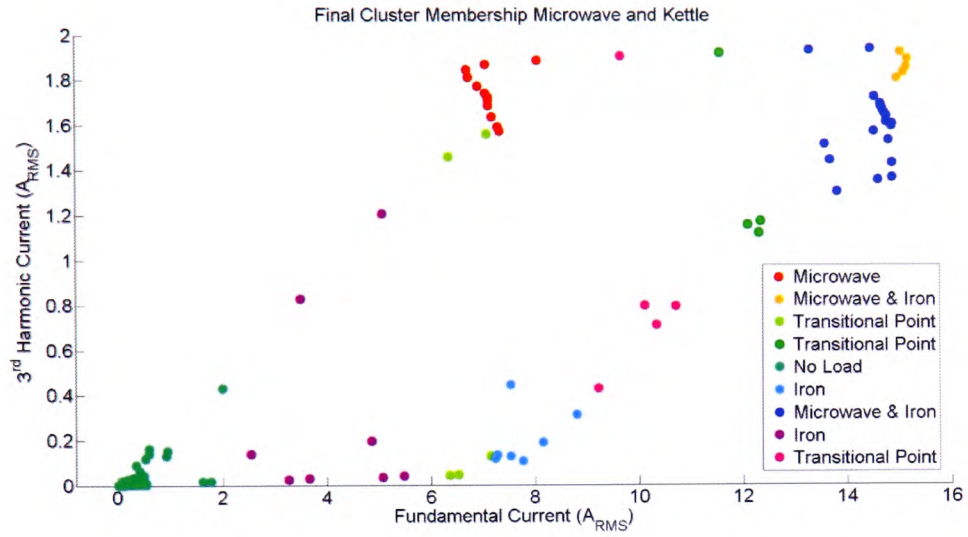

Figure 10-17 – Final Cluster Allocation

The final grouping of the loads can be seen within Figure 10-17.The plot shows that the final groups have been loosely captured, but due to the amount of transitions there were some groups that were split up into two adjacent groups' in the plot that should have been classed as one group, which can be seen by the microwave & iron group.

10.3.2. Further Clustering Analysis

To check the validity of the clustering analysis, further testing was conducted on a different dataset containing four different loads. The loads that were tested were a hair dryer, kettle, toaster and Microwave, which are all common loads found in the domestic environment.

The following plots show the frequency domain of the different loads, which are used for the clustering of the loads. Each of the plots shows a sample period of when the load was in the on state. It can be seen from Figure 10-18, Figure 10-19 and Figure 10-20 that the hair dryer, kettle and toaster respectively are all classed as linear loads as all the current is drawn in the fundamental frequency. Figure 10-21 shows the plot of the microwave in the frequency domain and can be seen to be non-linear as there is current in the third and fifth harmonic as well as the fundamental.

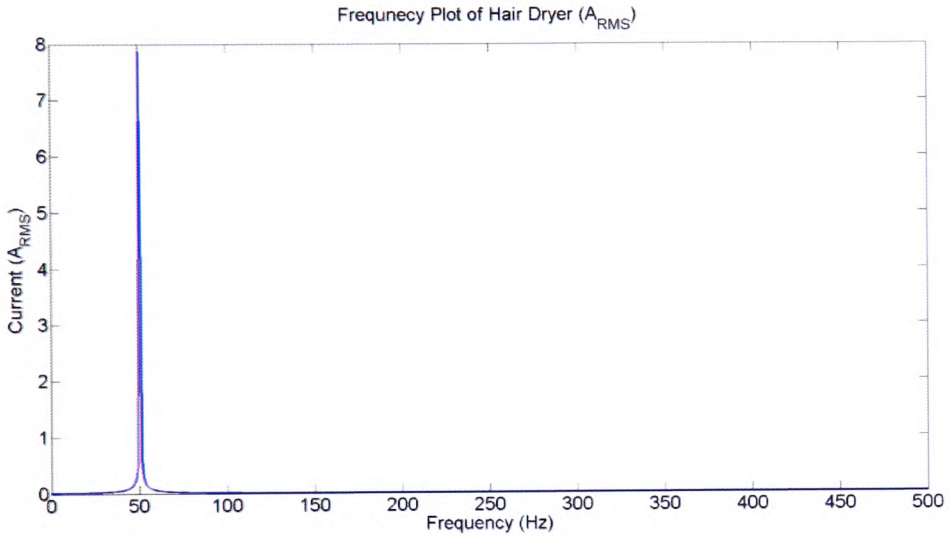

Figure 10-18 - Hair Dryer Frequency Plot

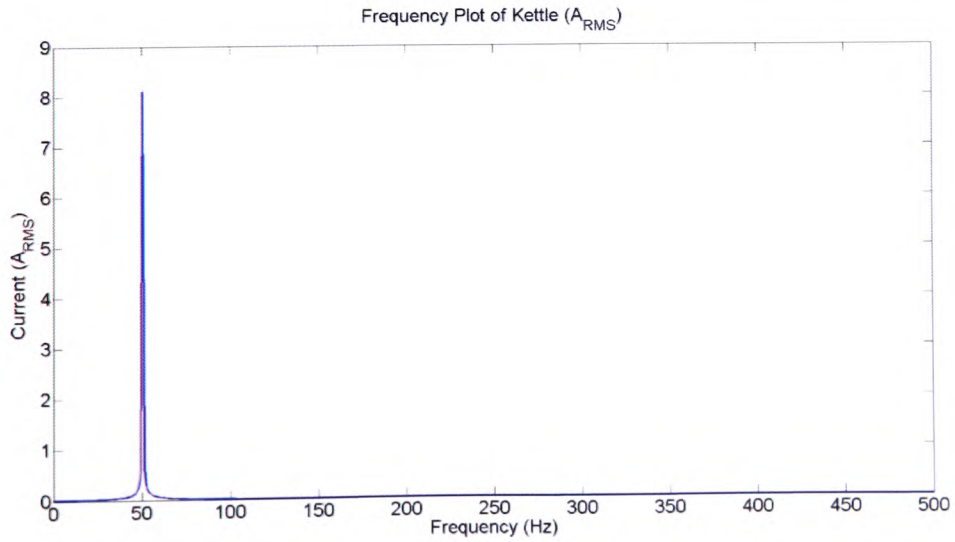

Figure 10-19 - Kettle Frequency Plot

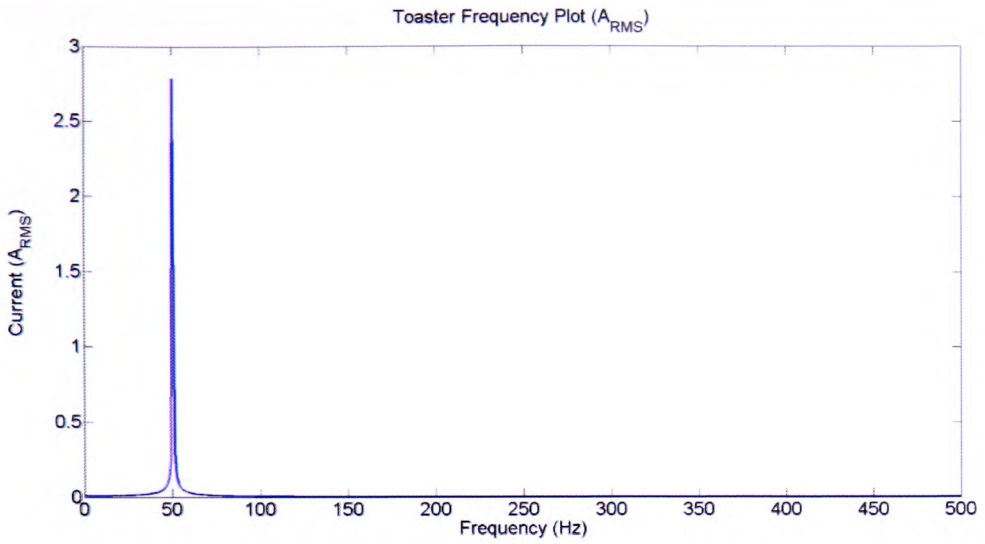

Figure 10-20 - Toaster Frequency Plot

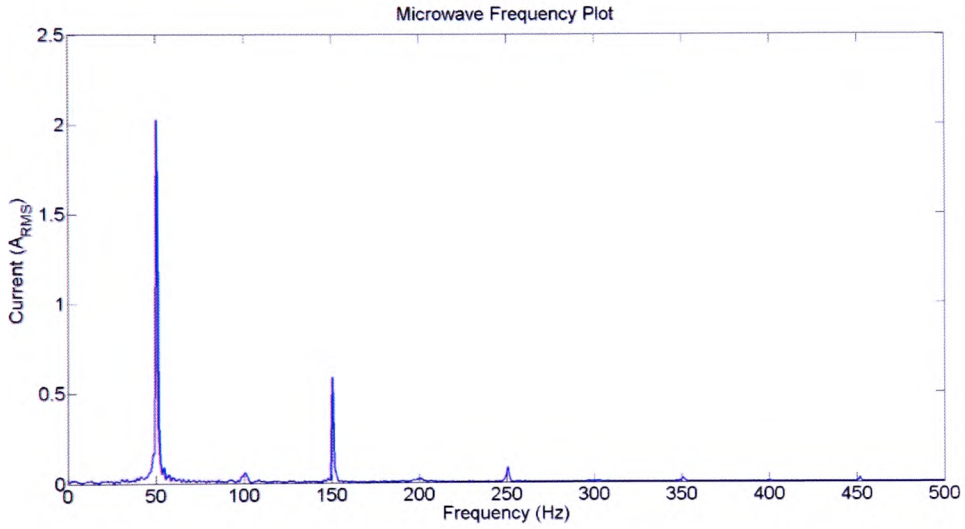

Figure 10-21 - Microwave Frequency Plot

The combined loads can be seen in the total load draw shown in Figure 10-22. Here the different loads can be seen, and the third harmonic can also be seen when the presence of the Microwave is being used. The variation in the fundamental and third harmonic is also shown and can be described as the variation in its operating conditions. One important characteristic of the plot is the turning on and the turning off of the different loads, which can be attributed to the initial loads starting up. This is especially prominent in the Microwave which can be seen in the third harmonic.

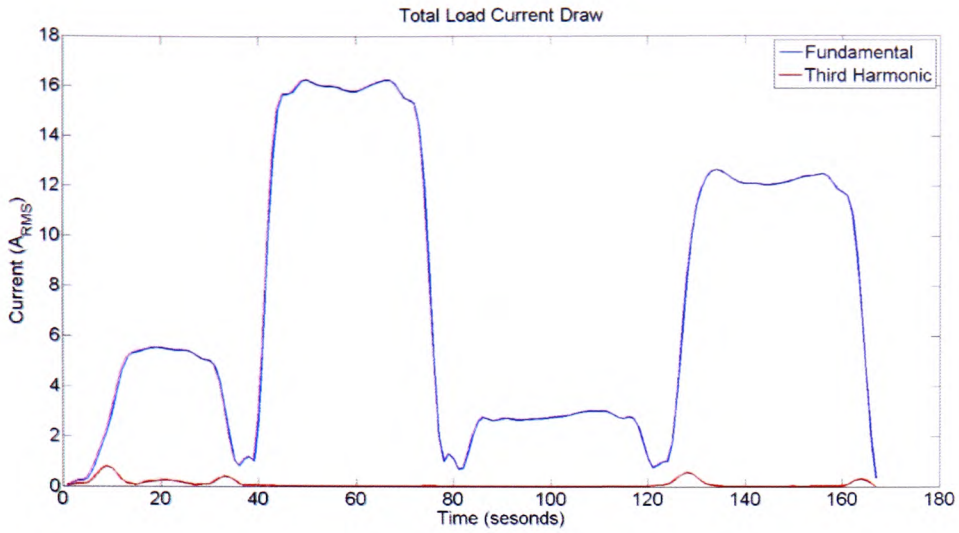

Figure 10-22 – Total Load Current, Fundamental and Third Harmonic

Each of the loads had been individually pre-processed to find out their operating conditions, and the final groups were found. These can be found in Figure 10-23. Out of the loads that needed to be described, the microwave showed to be the one with the most variation in its current draw in the third harmonic, which was shown in Figure 10-22. This variation was attributed to the increase in power as the microwave started up and continued through its operating cycle.

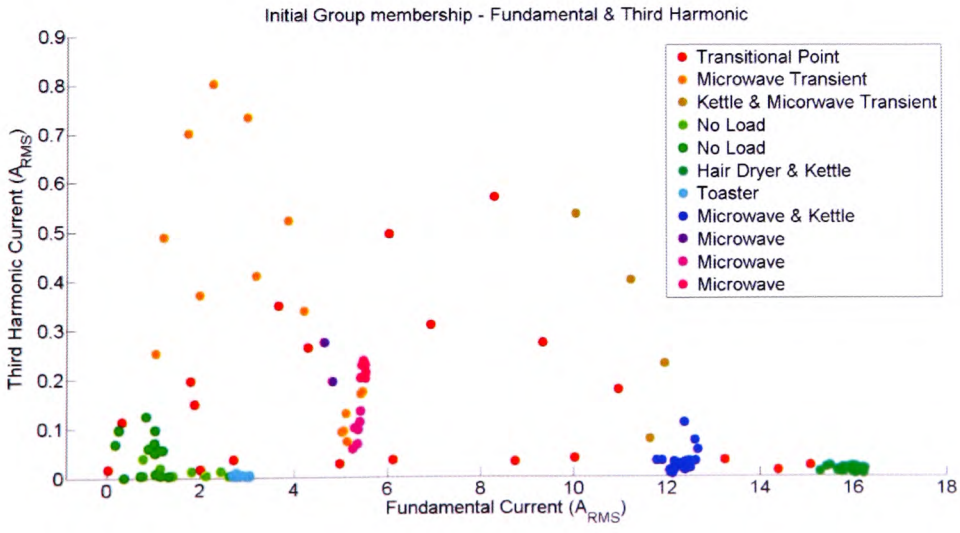

Figure 10-23 - Initial Group Membership, Fundamental & Third Harmonic

The other groups which contain the kettle, toaster and hair dryer can be seen to be separated from the other groups. There is only the issue around the transitional points that make the analysis more difficult and add a lot of noise to the final results. The recorded data was split into two mappers as per the algorithm, and the canopy centres were found. The results of mapper one and mapper two can be found in Figure 10-24 and Figure 10-25 respectively.

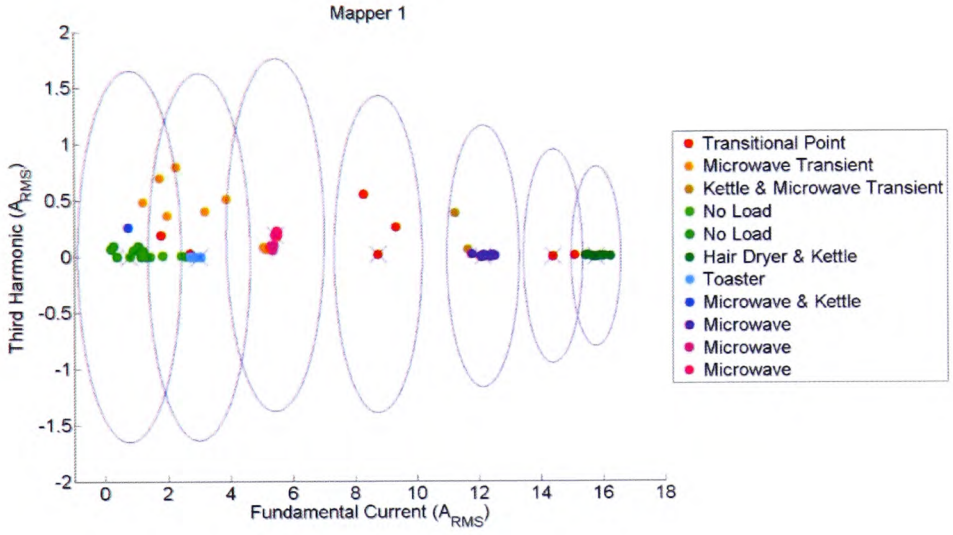

Figure 10-24 - Mapper 1

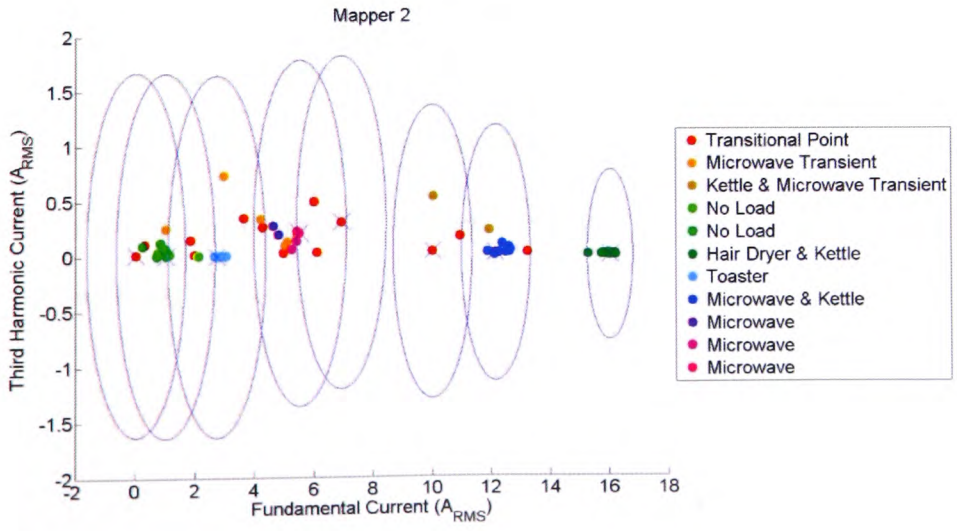

Figure 10-25 - Mapper 2

Combining the mappers shows the final canopy creations shown in Figure 10-26. The figure shows that the canopies have been created so that the respective groups are contained within the boundaries of the canopy. The final stage of the clustering was completed with the data points in the canopies only using k-means clustering.

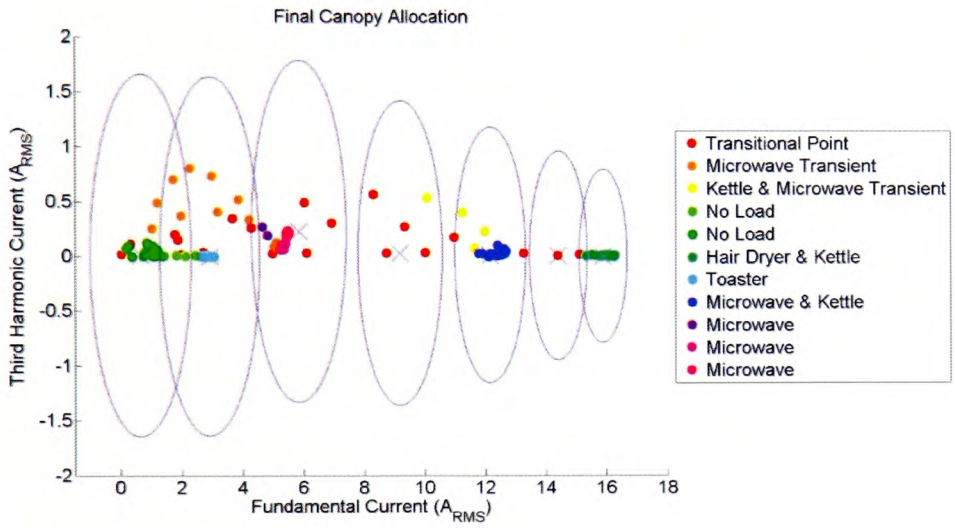

Figure 10-26 - Final Canopy Allocation

Figure 10-27 shows the final allocation of the data points to the clusters found in the dataset. The main loads can be seen to have been separated out by the canopy clustering. One of the interesting points to note is that the microwave transient points have now been either separated out into transient points or part of the main group. The transient of the microwave takes more than a few seconds to reach its maximum current before settling to its operating conditions, and as a result has not really featured in the results.

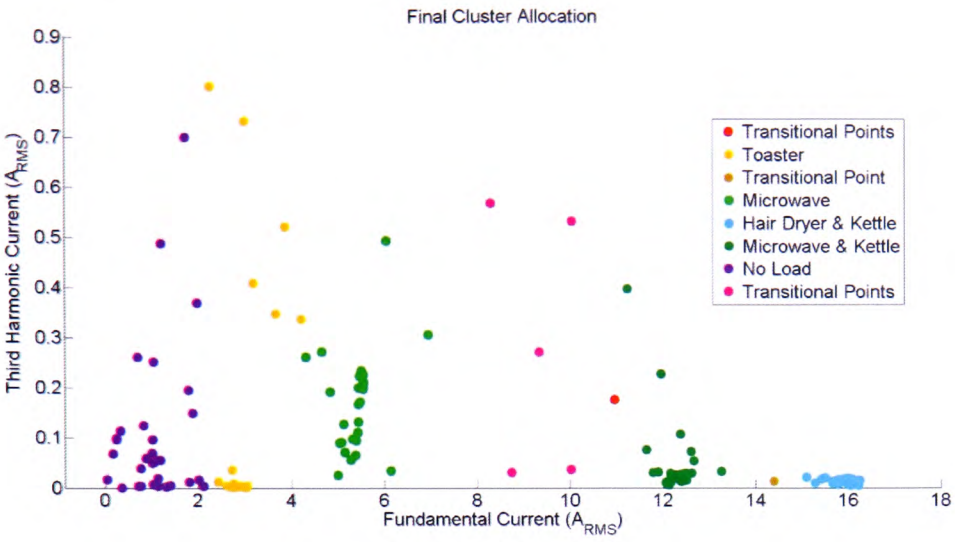

Figure 10-27 - Final Cluster Allocation

The larger energy consuming devices are easier to distinguish on the plot, and produce a cleaner, defined cluster. This is shown with the hair dryer and kettle, where it is shown to be further away from the noise of the no load conditions and the transitional points, making the larger loads easier to identify.

10.4. Comparing Resistive and Non-Linear Models

One of the main issues encountered with the previous results was that the models for the canopy relationship between the average of the group and the range or ratio are different. This became an issue when implementing the system into a real word environment, where the model could not be chosen through the processing period depending upon which loads were being used, as it was a Non-Intrusive load monitoring system, and this would be unknown.

To be able to identify a model that fits all purposes with the current technology becomes difficult when there are appliances such as the microwave that draws a residual current when the device is not actually operate. This residual current can be attributed to the fact that there are electronic devices on the casing of the appliance for the clock, which may also be back lit. This residual current could be seen within Figure 10-14

described by the group 1111, which effectively was the off state of the device. The issue of the current spread of the microwave can be seen within the other groups where there was large variation of current being drawn both in the fundamental and third harmonic, which will be attributed to the start-up of the device.

When the data was processed through the resistive load canopy relationship, the plot shown in Figure 10-28 can be seen to contain too many groups for the data set, with a total of 106 groups covering the total of 142 data points. The plot should display a total of four unique groups, along with the transitional points.

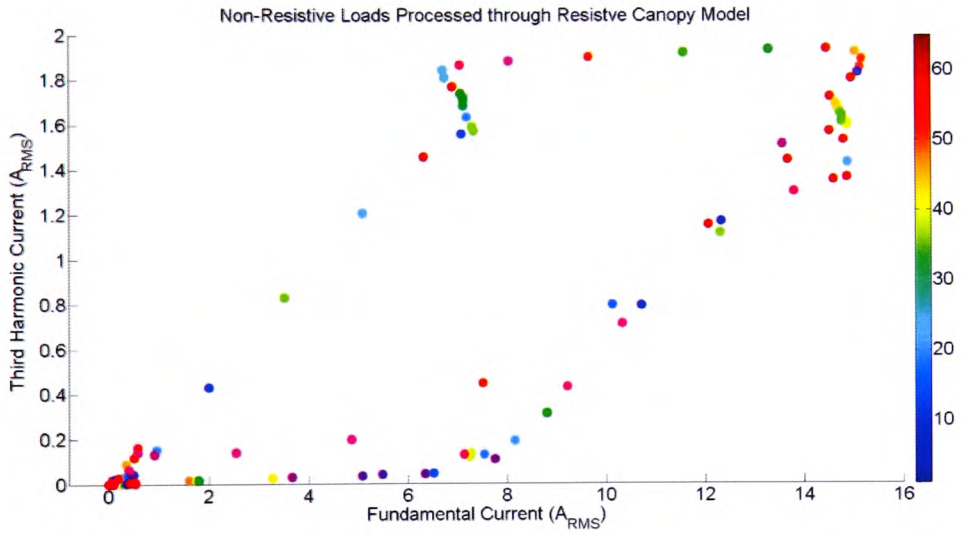

Figure 10-28 – Non-Linear Load with Resistive Canopy Model

Figure **10-28** shows that it is not possible to use the canopy plot described within the resistive load only model, as there is too much variation within the fundamental and third harmonic for it to be true, and this was due to the additional complexity of the operating characteristics of the microwave and most non-linear loads.

Alternatively the non-linear load canopy profile could be applied to the resistive loads for analysis. Figure 10-29 shows the output results of using the non-linear relationship between the canopy centre and the radius of the canopy. From this plot it

can be seen that the four groups have been correctly identified, with the transitional points scattered between, and have been allocated random groups, which shows that the use of the non-linear canopy model can be used to cluster resistive load datasets.

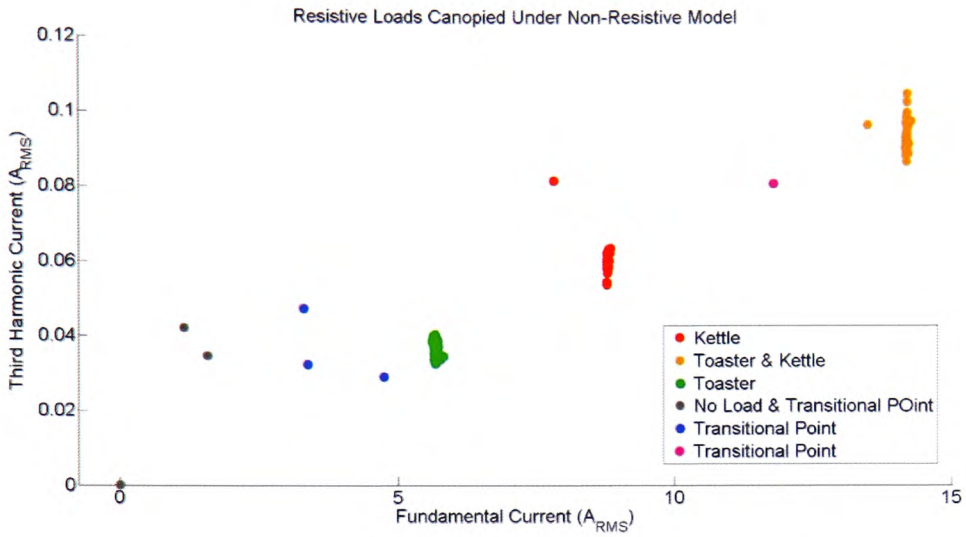

Figure 10-29 – Resistive Load with Non-Linear Canopy Model

The non-linear model has shown that it can be used to canopy cluster the two different datasets, and this model can be used for further testing on other loads which are typically seen within the domestic environment, such as a hair dryer and a lamp. The groups that require definition can be seen within Figure 10-30, and the details of which loads were on/off are shown in Table 10-6.

Hair Dryer	Lamp	Code
Off	Off	1111
Off	On	1121
On	Off	2111
On	On	2121

Table 10-6 – Lamp and Hair Dryer Group Definition

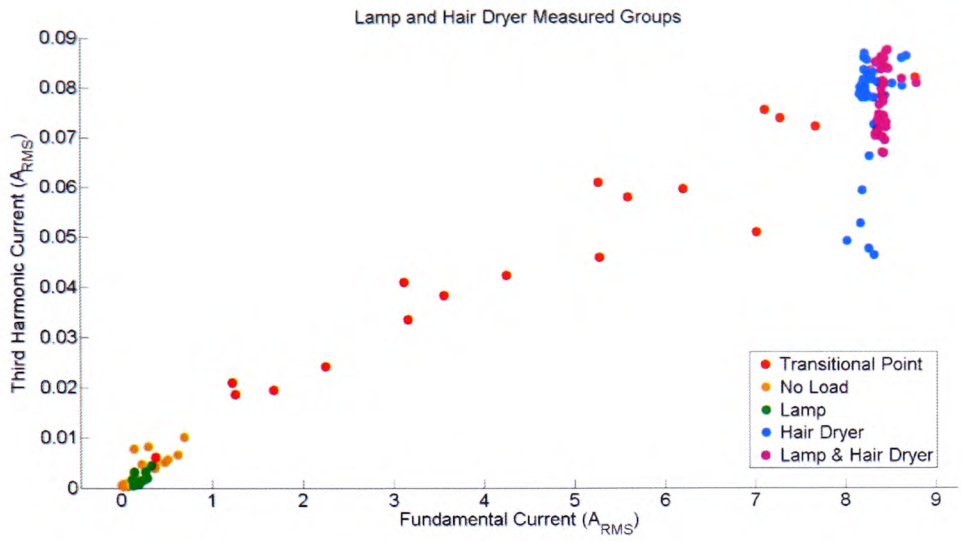

Figure 10-30 – Lamp and Hair Dryer Expected Groups

One of the initial observations that can be seen is that the lamp was drawing a very small amount of current, and this small amount of current form the initial plot showed that it was getting lost within the noise of the system, and could not be distinguished within the small scale of the current that it is drawing. The results showed that there were performance issues when it came to distinguishing the smaller loads from the larger loads when they were being used at the same time, and showed that there were limitations to the algorithm implementation when considering these smaller loads.

When zooming into the area of the lamp and no-load as seen within Figure 10-31, there was no clear distinction between the no-load state of the two loads, and the small amount of current that the lamp was consuming.

When considering the higher current values of the plot, the lamp and hair dryer, it could be seen that there was a very small area of distinction between when the lamp and when the hair dryer was on, to when the hair dryer is the solo energy consumer.

This was also emphasised by the fact that the group defined by the hair dryer alone, overlapped the group where the lamp was also present.

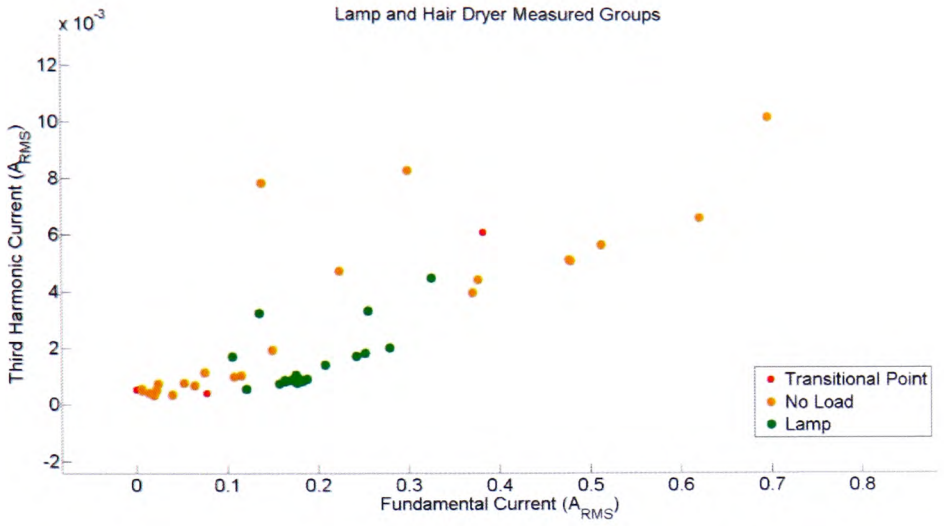

Figure 10-31 – Zoom of Lamp and No-Load

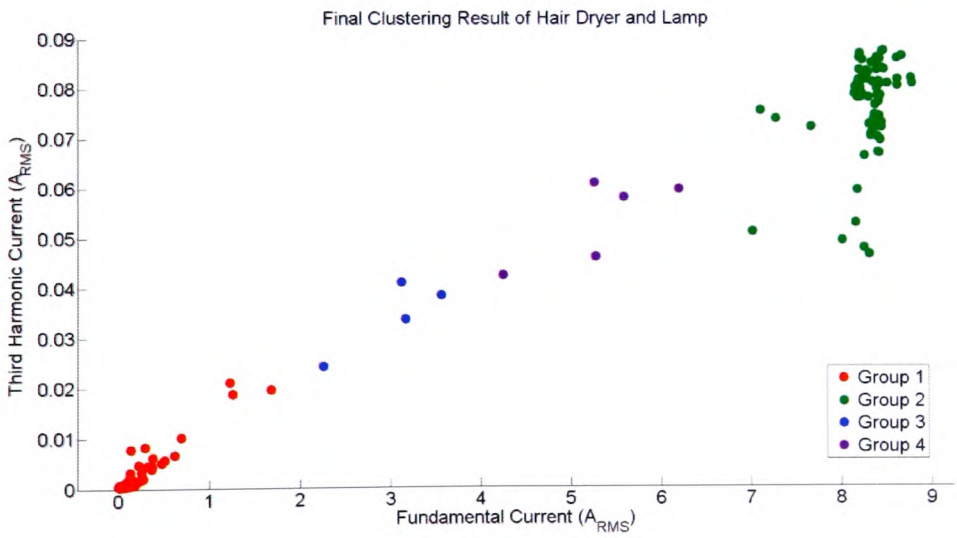

Figure 10-32 – Final Cluster Membership – Hair Dryer and Lamp

Figure 10-32 showed the final cluster membership when the dataset of the hair dryer and the lamp had been processed by the canopy clustering algorithm. The algorithm identified 4 groups, but out of those four groups there were only two actual

groups that are clearly definable; group one and group two. The other two groups that can be seen represent when the hair dryer is being turned on and off, and therefore could not be used as a group for load classification.

Group one showed that the no load conditions and the lamp on state had been identified as the same group. When considering when the hair dryer is on, again the overlap of the initial groups meant that the two separate groups representing the hair dryer and the lamp in both the on and off state had been combined as one group.

The experimentation showed that when smaller loads were considered with larger current drawing loads such as a hairdryer, the distinction between the points where the smaller load is present became difficult to separate out from the main dataset. When studied alone, there were clear distinctions between the on and off states of the individual load even at small currents. The larger loads operate over a range of current vales, and therefore the differences a small load make to the overall current draw could not be clearly identified as it would be operating within the natural variation of the current of the larger load.

10.5. Transitional Points

From the testing there is the constant occurrence of transitional points which are points that have been measured when a load has been turned on during the sample period, and is therefore not at the current operating condition or in no load status, and thus provides a rogue data points in the dataset.

The transitional points will have an effect on the final clustering of the loads and therefore needed to be addressed. The transitional points have been found during the pre-processing procedures and can be removed at this stage to test what the results would be without these data points.

10.6. Load Additions

The canopy clustering algorithm was used to separate out the groups of loads that were present within the premises. Due to this, further analysis was required for the classification of the individual loads within those groups. To be able to correctly identify the loads, the load interaction needs to be shown. When considering the loads contained within the resistive group, the main components that are used for classification are the fundamental components, as any residual current found within the higher harmonics was negligible.

When considering the non-linear loads the importance of the harmonics was shown with a significant portion of the current being within the higher harmonics. The ability to monitor and record the current harmonic components for the individual loads meant that the combination of loads could be determined by regression. An example of this can be seen within Figure 10-33, where the two individual loads are shown, with the total recorded load. When adding the two loads together, there is some slight variation between the recorded and the deduced load curves, but the general representation provides a good approximation.

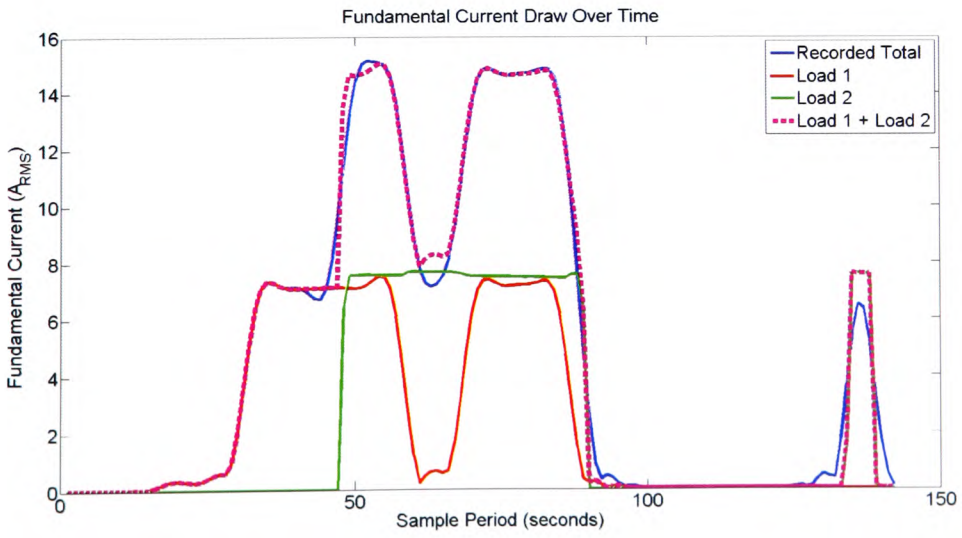

Figure 10-33 – Fundamental Current Draw Over Time

The main differences between the calculated and the recorded signals were due to the operating conditions of the loads, and how they interacted with one another with regards to the phase angles of the loads. This is where the phase angle of the appliance needed to be considered, as classifying the loads from the amplitude of the current alone did not contain all the relevant information. The output from the FFT function used for the creation of the data sets also allowed for the phase angle to be extracted. For this to be used in any meaningful way it needed to have a reference, which in this case was the voltage.

The calculations of the phase angle can be seen in Code Sample 10-1 within the text box. The first three lines of the code were used to determine the actual phase angle of the current with respect to the voltage. The alpha and alpha3 variables were used to determine the phase difference between the two load currents, which equated to beta within equation (9-1). This was required to calculate the amplitude of the resulting current represented by the variables c and c3.

```
IT1ang=Vang-I1ang;
IT2ang=Vang-I2ang;
ITTang=Vang-ITang;

alpha=abs(IT1ang(:,51)-IT2ang(:,51));
c=sqrt(I1d.FundHarmon.^2+I2d.FundHarmon.^2.+(2.*I1d.FundHarmon.*I2d
.FundHarmon.*alpha));
alpha3=abs(IT1ang(:,151)-IT2ang(:,151));
c3=sqrt(I1d.Harmon2.^2+I2d.Harmon2.^2.+(2.*I1d.Harmon2.*I2d.Harmon2
.*cos(alpha3)));
```

Code Sample 10-1 – Phase Angle Calculations

By looking at (9-1) the final load voltage of two loads may be found with the phase angle becoming a factor. Looking at Figure 10-34 which represents the resistive loads of the kettle and the toaster, the calculated values including the phase angles of the individual loads and the total load current was displayed. The figure showed that the

relationship between the individual loads added using both phase and magnitude information could be equated to that of the total load current measured.

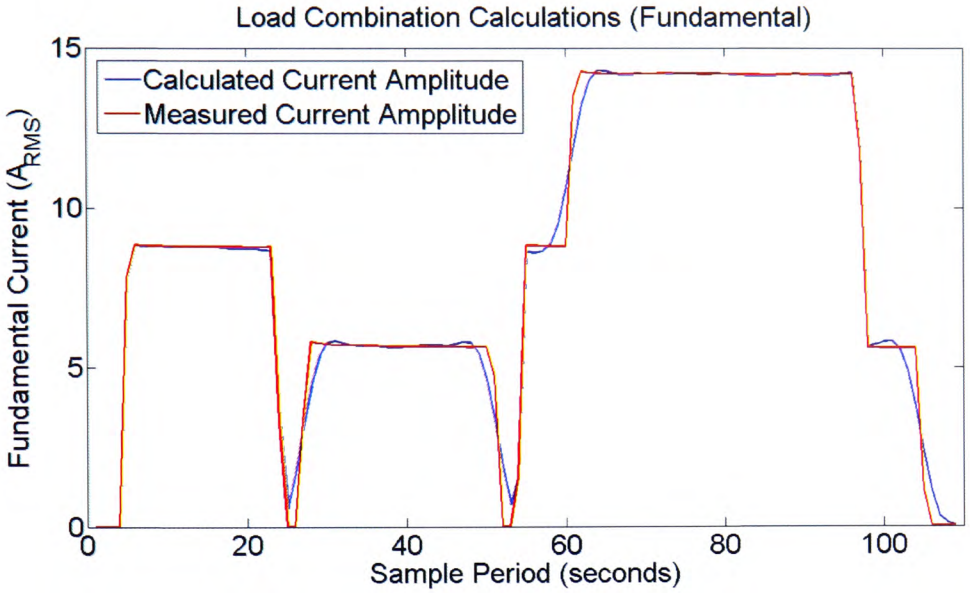

Figure 10-34 – Resistive Load Combination Calculation (Fundamental)

The figure shows that the known loads that were measured individually could be added using the trigonometric identities within equation (9-1). The knowledge of how the loads interact with one another could be used when trying to determine how the groups that were identified within the canopy clustering algorithm could be split into the individual loads.

In the case of the loads being of resistive load only, the fundamental alone would be enough for the classification of the load. The main issue with only considering the fundamental was that there was still some residual harmonic content contained within the higher harmonics, which could influence the final load classification of the groups of loads present. The same process was used for classifying the third harmonic data. Within this instance, the phase angle of the data was calculated from the relevant voltage harmonic, and this was used in the calculation of the combined load current, which can be seen within Figure 10-35.

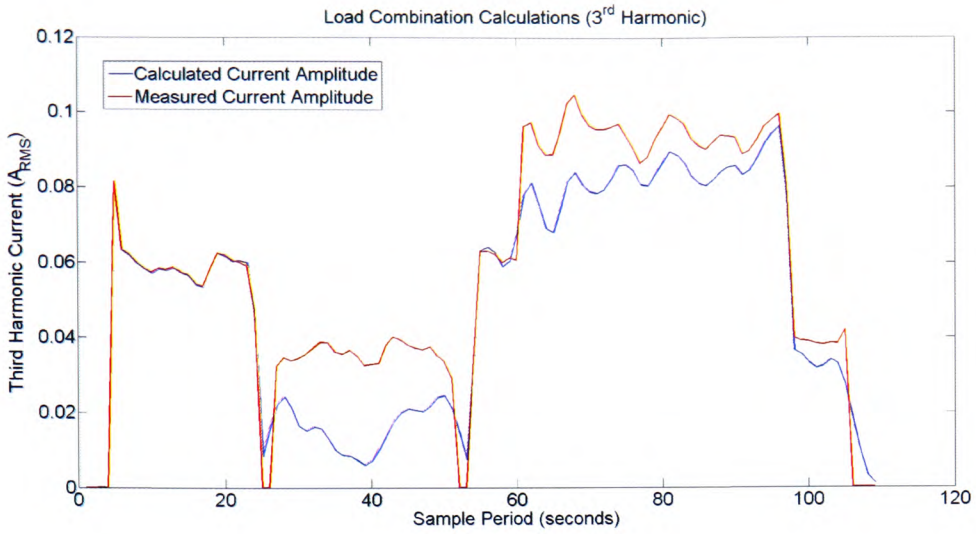

Figure 10-35 – Resistive Load Combination Calculation (Third Harmonic)

Figure 10-35 shows that there was some variation between the calculated and the measured current contained with the third harmonic, which was when the toaster was present. This difference between the measured and the calculated values could be due to noise within the system, but when looking at the actual scale of the third harmonic it is within the milliamp range which would be lost within system noise or when large non-linear loads were used. The kettle by comparison showed a good representation of the load when compared to the measured values.

When considering the information when there were non-linear loads present the same process of load combination calculations could be used. The main difference was that there was a greater influence within the higher harmonics, and the representation of the combination of loads within the higher harmonics was more important for the classification process.

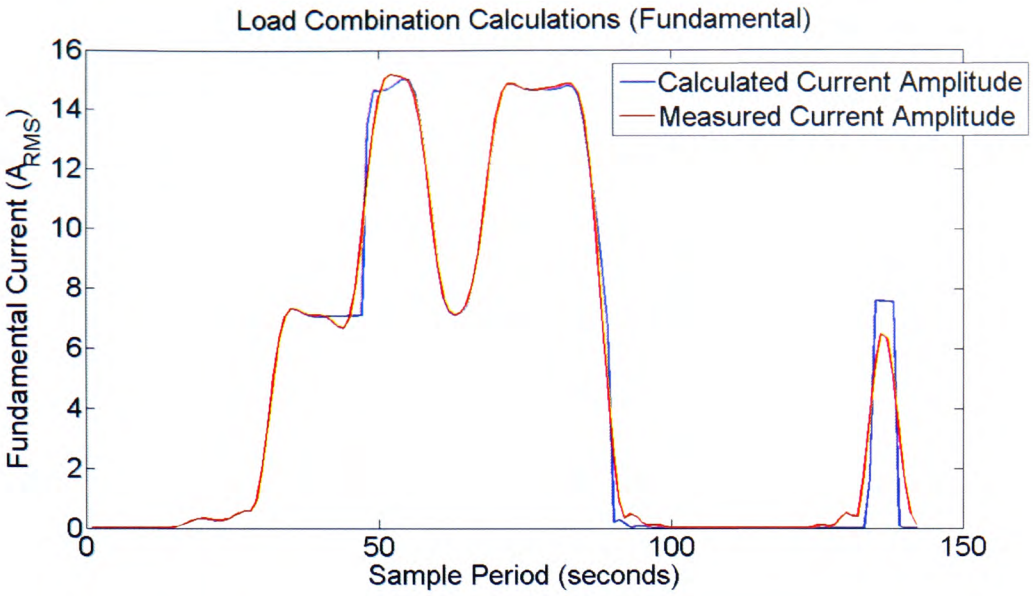

Figure 10-36 – Non-Linear Load Combination Calculation (Fundamental)

The plot shown in Figure 10-36 shows the representation of the calculated and measured values of the combination of an iron and a kettle. They can be clearly seen to be related, and actually shows the middle dip point where the iron alone was visually better than the plot shown in Figure 10-33 which was calculated without the phase angles.

The same method has been completed for the third harmonic, with the phase of the current calculated compared to that of the voltage. The results of the third harmonic analysis can be seen in Figure 10-37.

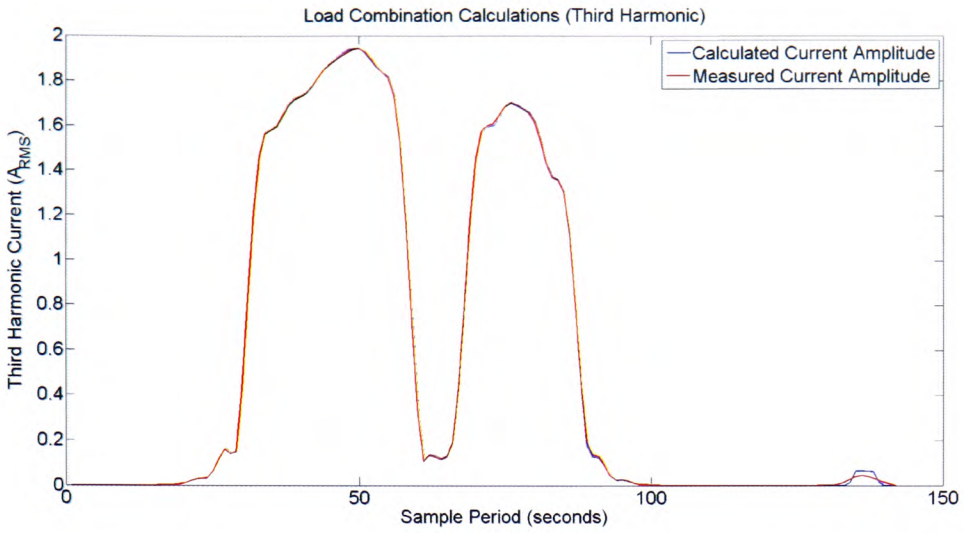

Figure 10-37 – Non -Linear Load Combination Calculation (Third Harmonic)

Comparing the third harmonic plot of the non-linear loads to that of the resistive, it can be seen that the representation of the calculated values provides a better fit within the non-linear plot. The calculated and the measured closely follow the pattern of the current draw over time very closely, and this was due to the fact that there was actual current being drawn within the higher harmonics.

One important observation to be made about the plot of the third harmonic to that of the fundamental is that the iron, which is resistive in it operation, is not shown to have any harmonic content within Figure 10-37 which is to be expected. This important characteristic can be used when determining which loads were present within the data as an absence of higher harmonics would indicate a resistive load. The opposite can be said when there is higher harmonic content, that the load would be considered as a non-linear type.

10.7. Implementation

The use of canopy clustering requires the frequency components of the loads to be found. The current implementation of this is carried out in MATLAB which will not be used in the final implementation. As a result the algorithm is required to be processed in

the confines of a metering device that is supplied and installed in the home. The meter therefore is required to be able to compute the FFT of the current drawn in the property.

Current technology is already available for the task and is currently being used for the development of a three-phase meter and test set at KIGG ltd. The processor is provided by Microchip numbered dsPIC33FJ128GP206A [109]. The processor provided by Microchip includes the DSP (Digital Signal Processing) at the hardware level that is required to compute a DFT (Digital Fourier Transform). The DFT is used instead of the FFT due to the processor conducting a DFT on 214 samples which cover three cycles of the 50Hz supply. The sample rate is not a power of two and therefore true FFT cannot be used, and therefore DFT is used in its place.

10.8. Computational Complexity

The computational complexity of the proposed algorithm can be compared to that of the k-means algorithm. The complexity of the k-means algorithm is determined by the amount of distance calculations required for the k-centres to converge, and is an iterative process. Each iteration is involves the calculation of the distance between each of the data points to the k-centres until convergence and its complexity can be described by (10-1) with N being the number of points in the dataset, l equal to the number of clusters and I the number of iterations.

$$\boldsymbol{T(N) = O(Nki)} \qquad (10\text{-}3)$$

When taking canopy clustering into consideration, the approach is different. The initial steps of the algorithm separate out the data into canopies and can be considered to have N operations. When in the canopies, k-means clustering is carried out in the individual canopy, with each canopy having one k-centre, thus equating k to one. The complexity computation will then be reliant upon the amount of canopies and the

number of data points per canopy, which are both related. With the extremes being taken into consideration if there is one canopy then there will be N points in the canopy, and conversely with N canopies, there will be one point per canopy. With this in mind the computation complexity of the distance calculations needed to reach the final ***N=O(CNCki)*** **(10-4)**, where C is the number of canopies.

$$T(N) = O(C\frac{N}{C}ki) \qquad (10\text{-}4)$$

With the assumption that there is one k-centre per canopy the final equation can ***N=O(Ni)*** **(10-5)**. The computation complexity of canopy clustering is reduced considerably and is dependent upon the number of data points in the dataset, where as traditional k-means, the calculations are increased by the amount of k-centres that are being found.

$$T(N) = O(Ni) \qquad (10\text{-}5)$$

10.9. Conclusions

The chapter has detailed the results from experimentation that has been completed from initial canopy clustering to defining the final groups that require classification. There were many different load types that require classification which have been touched on, and these could be separated into two main categories, resistive and non-linear loads.

Resistive loads were described as those which draw their current from only the fundamental frequency, and only contain resistive components. These loads were found typically within the domestic environment, and were representative of appliances such as kettles, irons and toasters. Due to the lack of any current draw within the higher harmonics, and the simple operation of the device being either on or off, their profiles could easily be created from the canopy plots shown earlier.

Non-linear loads added a further complication to the process of canopy clustering the loads. Unlike resistive loads, a large proportion of the current used by the appliance was drawn from higher harmonics, which distorted the current waveform. This added complexity of the appliance was attributed to the increased number of components used within the appliance, which would not just be resistive but contain both capacitive and inductive components. Also the presence of AC to DC power supplies which pull a non-sinusoidal current, added to the harmonic content within the current draw.

To further ensure that the results obtained from the canopy clustering algorithm were correct, each of the individual loads was analysed so that the load operating conditions were known. The operating conditions of the individual loads were combined for each of the experiments and the overall load operating conditions of the individual groups were clearly identifiable as seen within Figure 10-4 as an example. Each of the groups that require identification were shown within the four groups with both loads off, either load on only, and both loads on at the same instance in time. By monitoring the individual loads singularly and defining the groups, the relationships between the groups, and the groups parameters could be found.

By collating the information about the expected groups that were to be found through the canopy clustering algorithm, a model could be created that would be used for defining the size of the canopies used for the initial data segregation. Both the relationships between the resistive and non-linear loads were analysed within the results, and different approaches were found to provide better fits for the data.

With regards to the resistive loads, it was found that the relationship between the ratios of the range to the average of the group fitted a quadratic function as seen in Figure 10-5. This relationship was found using the information present for the dataset being analysed, and when considering the use of this approach, similar methods of

finding the relationship between the canopy centres and the boundaries would need to be found. The final fit showed the variation in the relationship of the canopy centres to the canopy boundaries, with the larger currents there was a smaller ratio needed to calculate the boundary. This is due to the increasing value of the current drawn, and the spread of the data in proportion to the average did not increase significantly compared to the average.

The canopy boundary relationship was found from the analysis and applied to the datasets with a solid fit being found, and the canopies could differentiate the loads from one another. Due to the nature of the canopy clustering there are some canopies that overlapped to cover the same group, which was to be expected, as the selection of the canopy centre is initially random, and therefore the optimal center may not be initially chosen.

By using mappers within the canopy algorithm, it was found that the fit of the canopy could be increased, while removing the need for excess overlapping canopies covering the same dataset. The mappers worked by separating the dataset into two or more sets, and then the results of the canopy analysis of the mappers were then combined to provide a better fit for the data. It was found that this process helped with the overall fit of the data. The process also had the advantage of scaling to large scale problems with its ability to be parallel processed.

The final group membership of the resistive load testing showed that the clustering methods managed to distinguish the individual groups, with a minimum accuracy of 85.71%, where the monitoring of the no load conditions was being grouped, and this lower accuracy was due to the transitional points that were found within the group, which are difficult to separate out of the main group.

Other groups showed that the canopy clustering managed to define a single load as two groups, which was due to the process of the canopies within the first stages of

the algorithm. These groups, while being defined as separate, contained all of the data points that are expected to be within them and further analysis would be required to combine these two groups into being the same load.

The non-linear loads were analysed also, but there were variations between the plots of the current draw within the fundamental and third harmonic domain. One of the main differences between the two types of loads were due to the residual current when loads were plugged in and the spread of the data was larger than that of the resistive load in both the fundamental and the third harmonic.

This difference in the operating conditions of the loads, lead to a different approach to be taken for determining the radius of the canopy clusters. Instead of using the ratio for the relationship, the actual range of the data was used and this is shown within Figure 10-15. The relationship has been described by a quadratic function.

The plots of the current content of the non-linear loads could be seen within Figure 10-17 and showed that there was a variation between the groups within both the fundamental and third harmonic. The variation with these two domains allowed the different groups to be analysed correctly and the groups can be clearly separated due to these variations.

The main difference between the two models that were used within the results section showed that there are some difficulties separating out loads which are of different types, such as resistive or non-linear. The non-linear loads show that there was a larger spread of the data points when the appliances are in the on state, and this larger range can be seen both in the fundamental and third harmonic. This difference in operating conditions between the loads meant that the relationship between group centres and the radius of the canopy needed to account for the spread of the data points.

When applying the non-linear canopy model to a further set of data such as the lamp and the hair dryer, the model was able to identify two separate groups. The actual

groups present should have been identified as 4 different groups as described in Table 10-6. The main issues that was found when trying to identify the different groups within the scatter plot was that the smaller difference in load current of the lamp when combined with the hair dryer was difficult to differentiate. This was due to the small amount of current that is being drawn by the lamp, and when comparing it to the operating conditions of the hair dryer, it was not large enough to be separated out.

Even though the lamp was clustered with the hair dryer in the canopy clustering process using the non-linear model, the model itself was good enough to use when monitoring larger energy consuming loads and showed that it could separate out the the larger loads that were being used over the sample period.

The non-linear model was applied to the resistive load data for comparison, and the results shown in Figure 10-29 show that each of the individual loads had been separated out into their relevant groups, and showed that the non-linear relationship between the canopy centre and the radius of the canopy could be used for both resistive and non-linear loads.

When it comes to load classification, there results have shown that it was possible to replicate the final groups by adding the individual groups together. The addition of the loads was completed using the trigonometric identities that describe how two sine waves interact and one important factor was that the phase angle of the individual loads was required, which could be calculated by using a reference phase such as the voltage.

Overlapping the plots of the measured total load and the calculated loads show a resemblance to the plots, and the errors between the two values was small. This error was small when considering the loads that were actually consuming current within the domain being measured. An example would be the third harmonic within the non-linear loads. When comparing this to the resistive loads, there was a smaller error within the

fundamental, but when the third harmonic was considered, the calculated values no longer matched with a close enough accuracy. This was due to the fact that the load current for the device was being consumed within the fundamental current and the amplitude of that current in the fundamental compared to the third harmonic was considerably larger.

The results showed that if the loads within the premises that were being monitored had an associated profile containing information of the current phase and amplitude, the groups of loads that were identified within the canopy clustering could be broken down into their individual appliance components. This was completed by analysing the individual loads and combining different combinations until the correct combination provided a match to the group of appliances being monitored. This method of classifying the loads could be done via a brute force method, or other methods of regression, which is out of the scope of this work.

The canopy clustering algorithm has shown that it is effective at disaggregating large loads which are typically found within the domestic environment. The key to using canopy clustering was the radius of the canopy, and this needed to be chosen to be large enough to cover the expected cluster and small enough to allow suitable distance between canopies so that multiple independent clusters were not grouped together. Analysing both resistive and non-linear loads gave a canopy relationship between the cluster centre and the canopy radius which showed different results. However it was found that by applying the non-linear model to that of the resistive data set acceptable results were found, and this relationship was used for the analysis. The ratios of the average to the range for the relationship for the canopy boundary were used so that the canopy boundaries could scale with the different load currents that were being consumed instead of using a static boundary for the canopies.

When further monitoring was conducted, there were some issues that were present when there was a small load included in the data, such as the low power lamp. This small load became lost amongst the noise of the system and the variable current draw of the high current drawing appliances such as the hair dryer. This meant that there was no clear way of separating out the smaller loads from the larger ones using canopy cluster, as effectively the two separate groups overlapped, and were found within the same canopy.

Canopy clustering used within NILM allowed for the separation of larger loads only in a premises, and by using the profiles of individual loads the groups identified within the canopies can be disaggregated into their component loads through methods of regression.

By analysing the larger loads within the premises there were still large energy savings to be made due to the amount of power that each of the loads consumes. By providing consumer feedback through the use of NILM, energy end users can make informed decisions on how, when and where their energy was being used and the use of canopy clustering within NILM could be effectively applied to aid in the feedback process.

Chapter 11. Conclusions and Further Work

11.1. Summary

The section draws the conclusions and discussed the future work of the research. The initial parts of the chapter reflect on the motivation, aims and objects of the research set out and how or if these have been met. The contribution to knowledge has been stated, followed by further work to be completed to take the concepts of the research forward to product development and wide scale implementation.

11.2. Motivation, Aims and Objectives Revisited

11.2.1. Motivation

The motivation of the project as describe within Chapter 1, was to develop methods whereby energy consumers can monitor their energy consumption within the home so that informed decision on energy savings can be made. By providing consumers with information on how and where energy is being used within the home can aid in that decision.

The way in which energy usage information was obtained could have been either through sub metering or methods such as NILM. Using NILM as opposed to sub metering meant that there would be less intrusion when installing such systems, whereas sub metering can become costly with the amount of hardware that would have to be installed by comparison.

By using NILM energy usage information could be fed back to the end user through means of the meter. To aid in the process of disaggregating the loads within the premises a design approach of implementing canopy clustering into NILM was considered. By implementing canopy clustering into the system, it allowed for greater efficiency for load monitoring, and offered the ability to be able to separate out groups

of loads that were present within the premises, without prior knowledge of the number of groups present.

11.2.2. Load Monitoring, Data Capture and Pre-processing

The load monitoring and the data capture used within the research has been described in detail within section 3.3 and Chapter 6. The process of monitoring the individual currents and the total currents has aided in the testing and validation of the overall results of the research.

By providing a strong methodology of data capture and signal monitoring, the process of capturing the data within NILM was completed using the same techniques for its final implementation. The sample rates of the system have been chosen to allow for high resolution of the harmonics that were considered, and were chosen to reduce aliasing within the lower harmonics. The third harmonic has been used thorough the research with the option of including the higher harmonics into the algorithm at a later stage should it be required for further information to be used within the load disaggregation process.

The methods used within the research for the data capture and the initial signal processing, have provided a platform for the NILM system to be built which is accurate and could be used in practical applications. The use of Fourier analysis has been adopted to provide as much detail about the current signals as possible, by providing phase and amplitude information for each of the harmonics present in the supply waveform.

The pre-processing of the data had been used for monitoring the individual loads that were being used for the testing. This was completed to ensure that the final results of the data could be validated and completed by monitoring the groups formed within the different relevant harmonics of the individual loads. This was used to identify the operating conditions of the loads. The processes were described in detail in section 7.6.

By pre-processing the data acquired for the individual loads, the actual operating conditions of the loads were found. These operating conditions varied depending upon the type of load being monitored and it was found that there could be a range of different operating conditions for the load such as a simple on/off device, or a device that had multiple operating conditions i.e. a hair dryer with two heat settings.

11.2.3. Load Disaggregation

Load disaggregation has been obtained within the research with the use of canopy clustering and load classification. This two stage approach has been described in sections Chapter 8 and Chapter 9 respectively. The initial separation of the loads and groups are discussed further within this section, with the load classification discussed within section 11.2.4

11.2.3.1. Canopy Clustering

The research has completed a literature review of the current methods used within NILM, and there were many techniques that had been used for monitoring loads such as steady state analysis to transient event detection, which were discussed in greater detail within the literature review. Each of the techniques aimed to separate out the data into profiles that had key identifiable characteristics for allocating the data to a specific load. One of the main issues with these techniques was that the initial separation of the data relied upon methods such as k-means clustering, which required some prior knowledge of the amount of loads or clusters that require segregation, and therefore becomes a limitation of the system.

By employing the use of canopy clustering, the amount of expected clusters was negated due to the operation of the process of canopy clustering and as a result, the algorithm could be used to separate out clusters from a data set. This separation did require some prior knowledge of the typical clusters that were to be found such as the

range of the data that the clusters were spread across, which have been shown within Chapter 10. The development of the canopy model required relationship between the average of the clusters to the radius to be found and needed to be selected so that the canopy created was large enough to cover the cluster. The previous knowledge of the relationship between the average and the canopy radius was easily found by monitoring known loads; whereas the number of clusters within the dataset were not known due the amount of different combinations of loads that could be on within the premises.

Taking these factors into consideration the use of canopy clustering lends itself well of the application of data segregation within NILM systems, and has been the choice for the primary technique used within load disaggregation.

11.2.3.2. Canopy Model Development

To be able to understand the relationship between the creation of canopies and the data clusters created through the frequency analysis of the current consumed by the loads, different experiments were conducted for resistive and non-linear loads.

The results of the different experiments showed that the relationships between the radius of the canopy required to cover the cluster and the average centre point of the fundamental value of the cluster were different for the resistive and non-linear loads. This difference between the two the sets of data can be seen within the section 10.2 and 10.3. The different models were created separately to show how the different load properties could influence the results.

The model created for the resistive load was able to segregate the data into its component parts, which allowed for the different loads to be identified within the fundamental and third harmonic. When applying the model created from the resistive loads to the non-linear dataset, the model did not fit and was found to be separating the dataset into a large amount of clusters. The reason for this was the non-linear loads showed that there was a larger spread of the data within both the fundamental and third

harmonic, which lead to the separation of the actual expected clusters into many canopies covering the data. The problem of the spread of the data within the different clusters could be seen when there were loads in the no-load condition such as the microwave, which was effectively not being used, but there was a residual current being drawn due to the on-board digital backlit display.

Due to the difference in the results of the resistive load model, another model was created for the non-linear load using the same methods described within section 10.2 and a new relationship between the average center point of the fundamental and the radius of the canopy. When applied to the non-linear loads, it was found to be able to separate out the data into the known groups.

Having two different relationship models for the different load types would not be viable within the scheme of NILM due to the fact that there would be no way of automatically separating out the different load types, as this goes against the principles of NILM. To alleviate this issue, the model from the non-linear load was applied to the resistive loads dataset, and the result was found to match well for the purposes of NILM.

When considering the actual application of the canopy clustering algorithm within the scope of NILM there were some issues that were found within the testing. The application of the non-linear load model for the canopy clustering meant that there were smaller loads that were present which were being lost within the noise of the loads within the off position, and therefore could not be singled out. This is due to the model being developed around the loads that have residual current that occupy a large current range within the fundamental domain. This poses issues when trying to monitor the magnitude of smaller devices within the premises.

The issue with the smaller loads could be seen when there were larger loads present. When considering the operating conditions of different loads, there were

variations in operating conditions due to inherent construction. These variations have been accounted for within the canopy model, but when considering the operating range of smaller loads, these will be significantly smaller than the variation of the larger loads, and are therefore merged within the larger loads cluster.

This issue surrounding smaller loads means that while canopy clustering works for disaggregating larger loads, the smaller loads will not be separated out when the larger loads are present. The research has shown that although the smaller loads cannot be separated out, when considering its implications within NILM and the purposes of NILM, the smaller loads do not contribute to the larger proportions of energy consumption, but large load management is where the real savings can be made.

The research has highlighted some key points within the application of canopy clustering within NILM. Canopy clustering can indeed be used successfully to separate out loads and groups of loads from measurements of the current draw of the premises. Canopy clustering lends itself to the monitoring of the larger loads within the premises alone which still provide feedback on energy usage to consumers on the large energy consuming devices, and can aid in the reduction of energy usage.

11.2.4. Load Classification

Load classification has been completed by considering loads found within the groups defined from the canopy clustering algorithm which were able to distinguish groups of loads that were present within the dataset. By analysing the loads individually from the onset, profiles of different appliance could be created providing information about phase and magnitude of the appliances within the frequency domain.

Section 9.2 has detailed how sine waves of the same frequency interact with one another and this information could be used to determine which loads were present within the groups defined by the canopy clustering algorithm.

The groups of loads can be disaggregated into their component loads using the information obtained through canopy clustering and comparing these to known load profiles. The trigonometric identities used within Chapter 9 have shown that as long as the phase and magnitude information are known for the individual loads, combinations of these loads can be found which equate the magnitude and phase of the final load cluster.

11.2.5. Hardware Implementation

The current hardware has been shown to be up to the task of implementing load monitoring using canopy clustering. The use of dedicated DSP processors has meant that the costly computation of FFT or DFT can be computed efficiently on the chip without the slowing down or the interruption of the metering's device primary task of energy billing.

11.3. Contribution to Knowledge

11.3.1. Non-Intrusive Load Monitoring using Canopy Clustering

The research has proposed new techniques for load disaggregation in NILM, which features the use of canopy clustering as the primary method of cluster segregation. Other research proposals have used different methods of NILM, which range from steady state, power on/off analysis, transient analysis and rules based algorithms.

A novel approach to NILM was theoretically developed to provide a capability to overcome the shortcomings found in a priori research. Models were created and verified and implementation success was shown through both simulation and empirical studies. The research uses canopy clustering as a means of initially disaggregating the groups of loads into their component clusters which differs to previous methods of NILM which try and classify the individual loads directly. Canopy clustering lends itself well to the disaggregation of loads due to its ability to separate out data into

overlapping canopies in a computationally efficient manner, without the need for prior knowledge of the amount of clusters that the final dataset is to include.

Previous methods of NILM such as on/off steady state analysis and transient analysis require that operating characterises of the loads are known, and a training period is carried out. By using canopy clustering in NILM, there is no requirement for specific training for the initial segregation of the data into the groups of loads present in the dataset. This method of NILM solves the inherent problems of the practical application of NILM in a domestic environment in that many loads will be in use at once. By approaching the problem with this in mind, groups of loads can be identified without any prior knowledge as to which loads was present in the premises. The groups of loads that are found from canopy clustering can then be further classified into their component loads, which is part of the further work to be conducted in this area of research.

The use of more computationally expensive clustering methods is still required for the final creation of the profile of the groups defined within the canopies but the advantage is that the clustering is carried out on the data contained within the canopy only and all other data points can be ignored, and to achieve this the use of k-means clustering was employed. The combination of canopy clustering and k-means clustering within the canopies allows for fast and efficient profile creation for the groups of loads to be identified within NILM, compared to traditional methods such as transient analysis, which requires the knowledge of start-up transient patterns to be known for the classification of individual loads.

By using canopy clustering in NILM, the training requirements and external input into the system is reduced, as the canopy clustering process initially separates out the data into smaller datasets that can then be pre-processed. From these groups that are defined, models of previously known loads can be used to identify the groups that are

present, with the knowledge that the loads present are the loads that are being used in the premises. This greatly reduces the training requirements of the NILM compared to the other methods such as rules-based, Neural Networks or steady state and transient analysis where it is important to have the load profiles of the individual loads for training the system. This allows for easier integration into the domestic environment as there are no requirements for lab training and makes the system user friendly from initial installation to every day consumer use.

11.4. Further Work

The research has identified that the use of canopy clustering within NILM provides an efficient and accurate method of separating out larger loads. There are some issues when the smaller loads that are present within the home, like bed side lamps, are taken into consideration which is due to the small amounts of current that is drawn. This can be lost within the current signal of the higher rated device, and the differences were hard to distinguish using canopy clustering.

Further analysis into other methods of disaggregating these smaller loads from the total load is required for the monitoring of smaller loads, which could involve greater in-depth analysis of the groups that are defined post canopy clustering.

The research has shown that canopy clustering can scale with parallel processing and the algorithms that have been developed for the testing of the theories that the research has been built, have been done purely in a successive algorithm. Greater efficiency in the clustering processes of the data can be created by employing the use of parallel processing technologies, and these could eventually be integrated in the process for commercial use as cost effective processors become available.

The processes of load disaggregation have been based on the grouping of loads which then require further analysis for the individual loads to be found. The research has shown that it is possible to find the loads that are present within the groups defined

by combining the profiles of the individual loads sing both magnitude and phase information of the loads obtained through Fourier analysis. Further work needs to be completed for compiling a database of common loads that are found within the domestic environment that will be monitored using NILM. Load profiles can be stored within an online database which could then be downloaded by smart meters that have internet connectivity. By providing these profiles, the implementation and training of the systems will become streamlined and therefore easier to integrate into widespread use.

The relationship between the different loads has been researched and to further commercialise the use of NILM within the wide scale market, further analysis into the final load disaggregation from the groups of loads that have been defined are required. This will be built on the use of the information obtained for loads within the load database, and could be accessed from networking the devices and remote storage of the information used in the algorithm.

References

[1] S. Drenker and A. Kader, "Nonintrusive monitoring of electric loads," *Computer Applications in Power, IEEE*, vol. 12, no. 4, pp. 47–51, 1999.

[2] L. Norford, S. Leeb, D. Luo, and S. Shaw, "Advanced electrical load monitoring: A wealth of information at low cost," *Diagnostics for Commercial Buildings: from Research to Practice, Pacific Energy Institute, San Francisco, CA*, 1999.

[3] S. R. Shaw, C. B. Abler, R. F. Lepard, D. Luo, S. B. Leeb, and L. K. Norford, "Instrumentation for high performance nonintrusive electrical load monitoring," *Journal of solar energy engineering*, vol. 120, p. 224, 1998.

[4] P. Shenavar and E. Farjah, "Novel embedded real-time NILM for electric loads disaggregating and diagnostic," in *EUROCON 2007 - The International Conference on "Computer as a Tool,"* 2007, pp. 1555–1560.

[5] M. Schwerndtner, "Digital measurement system for electricity meters," in *Metering and Tariffs for Energy Supply, Eighth International Conference on (Conf. Publ. No. 426)*, 1996, pp. 190–193.

[6] D. Bovankovich, "Submetering technology and applications," *Strategic planning for energy and the ldots*, 2005.

[7] Z. Xi, W. Xiong, S. Wang, and W. Huang, "Study on Electricity Sub-Metering System of Campus," in *Power and Energy Engineering Conference (APPEEC), 2011 Asia-Pacific*, 2011, pp. 1 –4.

[8] D. Infield and J. Short, "Potential for domestic dynamic demand-side management in the UK," *Power Engineering Society General Meeting, 2007. IEEE*, 2007.

[9] Y. Parag and S. Darby, "Consumer–supplier–government triangular relations: Rethinking the UK policy path for carbon emissions reduction from the UK residential sector," *Energy Policy*, vol. 37, no. 10, pp. 3984–3992, 2009.

[10] I. Mansouri, M. Newborough, and D. Probert, "Energy consumption in UK households: Impact of domestic electrical appliances," *Applied Energy*, vol. 54, no. 3, pp. 211–285, 1996.

[11] "Harmonic Solutions Co. Uk active & passive mitigation." [Online]. Available: http://www.harmonicsolutions.co.uk/. [Accessed: 04-Jun-2012].

[12] U. Dulleck and S. Kaufmann, "Do customer information programs reduce household electricity demand?—the Irish program," *Energy Policy*, vol. 32, no. 8, pp. 1025–1032, 2004.

[13] G. Strbac, "Demand side management: Benefits and challenges," *Energy Policy*, vol. 36, no. 12, pp. 4419–4426, 2008.

[14] S. Darby, "The effectiveness of feedback on energy consumption," *A Review for DEFRA of the Literature on Metering, Billing and direct Displays*, vol. 486, 2006.

[15] G. Barbose, C. Goldman, and B. Neenan, "A survey of utility experience with real time pricing," 2004.

[16] S. Darby, "Load Management at Home: Advantages and Drawbacks of some 'Active Demand Side' Options," *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, no. 227, pp. 9–17, 2012.

[17] S. Darby, "Why, What, When, How, Where and Who? Developing UK Policy on Metering, Billing and Energy Display Devices," *ACEEE Summer Study on Energy Efficiency in Buildings*, pp. 70–80, 2008.

[18] "Energy Billing and Metering: changing Customer Behaviour. Government response to a consultation." 2008.

[19] S. Deering, M. Newborough, and S. D. Probert, "Rescheduling electricity demands in domestic buildings," *Applied energy*, vol. 44, no. 1, pp. 1–62, 1993.

[20] T. M. I. Mahlia, M. F. M. Said, H. H. Masjuki, and M. R. Tamjis, "Cost-benefit analysis and emission reduction of lighting retrofits in residential sector," *Energy and buildings*, vol. 37, no. 6, pp. 573–578, 2005.
[21] G. Brandon and A. Lewis, "Reducing household energy consumption: a qualitative and quantitative field study," *Journal of Environmental Psychology*, vol. 19, pp. 75–86, 1999.
[22] R. Shaw and M. Attree, "The value of reducing distribution losses by domestic load-shifting: a network perspective," *Energy Policy*, vol. 37, no. 8, pp. 3159–3167, 2009.
[23] R. Shaw, M. Attree, T. Jackson, and M. Kay, "The value of reducing distribution losses by domestic load-shifting: a network perspective," *Energy Policy*, vol. 37, no. 8, pp. 3159–3167, 2009.
[24] E. Hughes, J. Hiley, K. Brown, and I. M. Smith, *Hughes electrical and electronic technology*. Pearson Education, 2008.
[25] B. Neenan and R. C. Hemphill, "Societal benefits of smart metering investments," *The electricity journal*, vol. 21, no. 8, pp. 32–45, 2008.
[26] "On energy end-use efficiency and energy services 2006/32/EC," 05-Apr-2012. [Online]. Available: http://europa.eu/legislation_summaries/energy/energy_efficiency/l27057_en.htm. [Accessed: 12-Mar-2012].
[27] S. Darby, "Why, What, When, How, Where and Who? Developing UK Policy on Metering, Billing and Energy Display Devices Why Develop Energy Feedback and Smart Metering?," *Buildings*, pp. 70–81, 2008.
[28] "Meter, Meter on the Wall: Giving & Taking from a Smarter Grid." [Online]. Available: http://www.triplepundit.com/2007/12/meter-meter-on-the-wall-giving-taking-from-a-smarter-grid/. [Accessed: 04-Jun-2012].
[29] G. Wood and M. Newborough, "Dynamic energy-consumption indicators for domestic appliances: environment, behaviour and design," *Energy and Buildings*, vol. 35, no. 8, pp. 821–841, 2003.
[30] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.
[31] M. Bergés and K. Shao, "Classifying Electrical Appliance State Transitions from Power Metrics Time-Series," *Poster Session of Machine Learning*, 2008.
[32] A. Meier, L. Rainer, and S. Greenberg, "Miscellaneous electrical energy use in homes," *Energy*, vol. 17, no. 5, pp. 509–518, 1992.
[33] A. I. Cole and A. Albicki, "Algorithm for nonintrusive identification of residential appliances," in *Circuits and Systems, 1998. ISCAS'98. Proceedings of the 1998 IEEE International Symposium on*, 1998, vol. 3, pp. 338–341.
[34] R. Singh and A. Singh, "Energy loss due to harmonics in residential campus—A case study," in *Universities Power Engineering Conference (UPEC), 2010 45th International*, 2010, pp. 1–6.
[35] I. Daut, H. S. Syafruddin, R. Ali, M. Samila, and H. Haziah, "The effects of harmonic components on transformer losses of sinusoidal source supplying non-linear loads," *American Journal of Applied Sciences*, vol. 3, no. 12, pp. 2131–2133, 2006.
[36] S. B. Leeb, S. R. Shaw, and J. L. Kirtley Jr, "Transient event detection in spectral envelope estimates fornonintrusive load monitoring," *IEEE Transactions on Power Delivery*, vol. 10, no. 3, pp. 1200–1210, 1995.
[37] S. Giri and M. Berges, "A study on the feasibility of automated data labeling and training using an EMF sensor in NILM platforms," in *Proceedings of the 2012 International EG-ICE Workshop on Intelligent Computing*, Herrsching, Germany, 2012.

[38] J. T. Powers, B. Margossian, B. A. Smith, Q. C. Inc, and C. A. Berkeley, "Using a rule-based algorithm to disaggregate end-use load profiles from premise-level data," *IEEE Computer Applications in Power*, vol. 4, no. 2, pp. 42–47, 1991.

[39] M. C. . Nguyen and W. J. Lee, "An approach to enhance the harmonic sources identification process," in *Industrial and Commercial Power Systems Technical Conference, 2000. Conference Record. Papers Presented at the 2000 Annual Meeting. 2000 IEEE*, 2000, pp. 127–132.

[40] U. Grasselli, R. Lamedica, and A. Prudenzi, "Time-varying harmonics of single-phase nonlinear appliances," in *Power Engineering Society Winter Meeting, 2002. IEEE*, 2002, vol. 2, pp. 1066–1071.

[41] F. Sultanem, "Using appliance signatures for monitoring residential loads at meter panel level," *Power Delivery, IEEE Transactions on*, vol. 6, no. 4, pp. 1380–1385, 1991.

[42] R. Dwyer, A. K. Khan, M. Mcgranaghan, L. Tang, R. K. Mccluskey, R. Sung, and T. Houy, "Evaluation of harmonic impacts from compact fluorescent lights on distribution systems," *Power Systems, IEEE Transactions on*, vol. 10, no. 4, pp. 1772 –1779, Nov. 1995.

[43] D. Srinivasan, W. Ng, and A. Liew, "Neural-network-based signature recognition for harmonic source identification," *IEEE Transactions on Power Delivery*, vol. 21, no. 1, pp. 398–405, 2006.

[44] "Encefalus." [Online]. Available: http://encefalus.com/. [Accessed: 12-Mar-2012].

[45] J. G. Roos, I. E. Lane, E. C. Botha, and G. P. Hancke, "Using Neural networks for non-intrusive monitoring of industrial electrical loads," in *Instrumentation and Measurement Technology Conference, 1994. IMTC/94. Conference Proceedings. 10th Anniversary. Advanced Technologies in I & M., 1994 IEEE*, 1994, pp. 1115–1118.

[46] P. K. Dash, D. P. Swain, B. R. Mishra, and S. Rahman, "Power quality assessment using an adaptive neural network," in *Power Electronics, Drives and Energy Systems for Industrial Growth, 1996., Proceedings of the 1996 International Conference on*, 1996, vol. 2, pp. 770–775.

[47] R. K. Hartana and G. G. Richards, "Constrained neural network-based identification of harmonic sources," *Industry Applications, IEEE Transactions on*, vol. 29, no. 1, pp. 202–208, 1993.

[48] W. M. Grady and R. J. Gilleskie, "Harmonics and how they relate to power factor," *Proceedings of PQA93*, 1993.

[49] M. Rukonuzzaman and M. Nakaoka, "Magnitude and phase determination of harmonic currents by adaptive learning back-propagation neural network," in *Power Electronics and Drive Systems, 1999. PEDS '99. Proceedings of the IEEE 1999 International Conference on*, 1999, vol. 2, pp. 1168 –1173 vol.2.

[50] M. Berges, E. Goldman, H. S. Matthews, and L. Soibelman, "Training Load Monitoring Algorithms on Highly Sub-Metered Home Electricity Consumption Data," *Tsinghua Science & Technology*, vol. 13, pp. 406–411, 2008.

[51] L. F. Wood, *Training neural networks*. Google Patents, 1990.

[52] M. Berges, "Building Commissioning as an Opportunity for Training Non-Intrusive Load Monitoring Algorithms," *Proc. of the 16th ldots*, 2010.

[53] M. L. Marceau and R. Zmeureanu, "Nonintrusive load disaggregation computer program to estimate the energy consumption of major end uses in residential buildings," *Energy Conversion and Management*, vol. 41, no. 13, pp. 1389–1403, Sep. 2000.

[54] S. Frank, L. G. Polese, E. Rader, M. Sheppy, and J. Smith, "Extracting Operating Modes from Building Electrical Load Data," in *Green Technologies Conference (IEEE-Green), 2011 IEEE*, 2011, pp. 1 –6.

[55] C. Laughman, K. Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong, "Power signature analysis," *IEEE Power & Energy Magazine*, 2003.

[56] S. R. Shaw, S. B. Leeb, L. K. Norford, and R. W. Cox, "Nonintrusive load monitoring and diagnostics in power systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 7, pp. 1445–1454, 2008.

[57] W. Wichakool, A. T. Avestruz, R. W. Cox, and S. B. Leeb, "Resolving power consumption of variable power electronic loads using nonintrusive monitoring," in *Power Electronics Specialists Conference, 2007. PESC 2007. IEEE*, 2007, pp. 2765–2771.

[58] C. E. Reeg and T. J. Overbye, "Algorithm development for non-intrusive load monitoring for verification and diagnostics," in *North American Power Symposium (NAPS), 2010*, pp. 1–5.

[59] M. S. Tsai and Y. H. Lin, "Development of a non-intrusive monitoring technique for appliance' identification in electricity energy management," in *Advanced Power System Automation and Protection (APAP), 2011 International Conference on*, 2011, vol. 1, pp. 108 –113.

[60] A. Cole and A. Albicki, "Nonintrusive identification of electrical loads in a three-phase environment based on harmonic content," in *Instrumentation and Measurement Technology Conference, 2000. IMTC 2000. Proceedings of the 17th IEEE*, 2000, vol. 1, pp. 24 –29 vol.1.

[61] M. Akbar and D. Z. A. Khan, "Modified Nonintrusive Appliance Load Monitoring For Nonlinear Devices," in *Multitopic Conference, 2007. INMIC 2007. IEEE International*, 2007, pp. 1 –5.

[62] L. K. Norford and S. B. Leeb, "Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms," *Energy & Buildings*, vol. 24, no. 1, pp. 51–64, 1996.

[63] H. Y. Lam, G. S. K. Fung, and W. K. Lee, "A Novel Method to Construct Taxonomy Electrical Appliances Based on Load Signaturesof," *Consumer Electronics, IEEE Transactions on*, vol. 53, no. 2, pp. 653 –660, May 2007.

[64] S. B. Leeb and J. L. Kirtley Jr, "A multiscale transient event detector for nonintrusive load monitoring," in *Industrial Electronics, Control, and Instrumentation, 1993. Proceedings of the IECON'93., International Conference on*, 1993, pp. 354–359.

[65] U. A. Khan, S. B. Leeb, and M. C. Lee, "A multiprocessor for transient event detection," *Power Delivery, IEEE Transactions on*, vol. 12, no. 1, pp. 51 –60, Jan. 1997.

[66] A. I. Cole and A. Albicki, "Data extraction for effective non-intrusive identification of residential power loads," in *Instrumentation and Measurement Technology Conference, 1998. IMTC/98. Conference Proceedings. IEEE*, 1998, vol. 2, pp. 812 –815 vol.2.

[67] L. Farinaccio and R. Zmeureanu, "Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses," *Energy and Buildings*, vol. 30, no. 3, pp. 245–259, 1999.

[68] "File:Stromwandler Zeichnung.svg - Wikipedia, the free encyclopedia," 2012. [Online]. Available: http://en.wikipedia.org/wiki/File:Stromwandler_Zeichnung.svg. [Accessed: 01-Apr-2013].

[69] W. F. Ray and R. M. Davis, "Wide bandwidth Rogowski current transducers: Part 1–The Rogowski coil," *EPE Journal*, vol. 3, no. 1, pp. 51–59, 1993.

[70] M. Faifer and R. Ottoboni, "An Electronic Current Transformer Based on Rogowski Coil," in *Instrumentation and Measurement Technology*, 2008, pp. 1554–1559.

[71] W. F. Ray, C. R. Hewson, J. M. Metcalfe, P. E. M. Ltd, and U. K. Nottingham, "High frequency effects in current measurement using Rogowski coils," in *Power Electronics and Applications*, 2005, p. 9.
[72] W. F. Ray and R. M. Davis, "Wide bandwidth Rogowski current transducers: Part 2–The Integrator," *EpE Journal*, vol. 3, no. 2, pp. 116–122, 1993.
[73] "File:Rogowsky coil.png - Wikimedia Commons," 2012. [Online]. Available: http://commons.wikimedia.org/wiki/File:Rogowsky_coil.png. [Accessed: 01-Apr-2013].
[74] W. F. Ray and C. R. Hewson, "High performance Rogowski current transducers," presented at the Industry Applications Conference, 2000. Conference Record of the 2000 IEEE, 2000, vol. 5, pp. 3083 – 3090.
[75] W. F. Ray, "Rogowski transducers for high bandwidth high current measurement," in *Low Frequency Power Measurement and Analysis (Digest No. 1994/203), IEE Colloquium on*, 1994, pp. 10–1.
[76] D. A. Ward and J. L. . Exon, "Using Rogowski coils for transient current measurements," *Engineering science and education journal*, vol. 2, no. 3, pp. 105–113, 1993.
[77] C. Qing, L. Hong-bin, C. Xiao, D. Yin, and L. Yan-bin, "PCB Rogowski Sensor Designs for Plasma Current Measurement," in *Fusion Engineering, 2007. SOFE 2007. 2007 IEEE 22nd Symposium on*, 2007, pp. 1–4.
[78] D. G. Pellinen, M. S. Di Capua, S. E. Sampayan, H. Gerbracht, and M. Wang, "Rogowski coil for measuring fast, high-level pulsed currents," *Review of Scientific Instruments*, vol. 51, no. 11, pp. 1535–1540, 1980.
[79] J. Cooper, "On the high-frequency response of a Rogowski coil," *Journal of Nuclear Energy Part C Plasma Physics Accelerators Thermonuclear Research*, vol. 5, no. 5, pp. 285–289, 1963.
[80] W. Ray and R. Davis, "High frequency improvements in wide bandwidth Rogowski current transducers," in *Proc. of EPE'99*, 1999.
[81] C. Hewson, W. Ray, and R. Davis, "Verification of Rogowski current transducer's ability to measure fast switching transients," in *Applied Power Electronics Conference and Exposition, 2006. APEC'06. Twenty-First Annual IEEE*, 2006, p. 7–pp.
[82] C. Hewson and W. R. Ray, "The effect of electrostatic screening of Rogowski coils designed for wide-bandwidth current measurement in power electronic applications," in *Power Electronics Specialists Conference, 2004. PESC 04. 2004 IEEE 35th Annual*, 2004, vol. 2, pp. 1143– 1148 Vol.2.
[83] W. F. Ray, "The use of Rogowski coils for low amplitude current waveform measurement," in *Measurement Techniques for Power Electronics, IEE Colloquium on*, 1992, pp. 4–1.
[84] J. A. J. Pettinga and J. Siersema, "A polyphase 500 kA current measuring system with Rogowski coils," *Electric Power Applications, IEE Proceedings B*, vol. 130, no. 5, pp. 360–363, 1983.
[85] N. Chen, K. L. Chen, and Y. P. Tsai, "Replacing current transformers with power current microsensors based on hall ICs without iron cores," in *Applied Measurements For Power Systems (AMPS), 2010 IEEE International Workshop on*, 2010, pp. 22–26.
[86] S. A. Dyer, *Survey of instrumentation and measurement*. John Wiley & Sons, 2001.
[87] "BMC Messsysteme GmbH - PCI-BASEII: PCI Measurement Card for Measuring Modules MAD/MDA/MCAN." [Online]. Available: http://www.bmcm.de/us/pr-pci-base.html. [Accessed: 29-Nov-2011].

[88] "BMC Messsysteme GmbH - MAD16f: Analog Input Module for PCI-BASEII, PCIe-BASE." [Online]. Available: http://www.bmcm.de/us/pr-mad16f.html. [Accessed: 29-Nov-2011].
[89] "BMC Messsysteme GmbH - USB-AD: USB Measuring System." [Online]. Available: http://www.bmcm.de/us/pr-usb-ad.html. [Accessed: 29-Nov-2011].
[90] "MathWorks United Kingdom - MATLAB - The Language Of Technical Computing." [Online]. Available: http://www.mathworks.co.uk/products/matlab/. [Accessed: 29-Nov-2011].
[91] "MathWorks United Kingdom - Statistics Toolbox - MATLAB." [Online]. Available: http://www.mathworks.co.uk/products/statistics/. [Accessed: 29-Nov-2011].
[92] "Allegro | Products | ACS756 Fully Integrated, Hall Effect-Based Linear Current Sensor IC with 3 kVRMS Voltage Isolation and a Low-Resistance Current Conductor." [Online]. Available: http://www.allegromicro.com/en/Products/Part_Numbers/0756/index.asp. [Accessed: 29-Nov-2011].
[93] S. Harrison and A. Manson, *SRSM and Beyond Smart Meter Specification*. Energy Retail Association, 2002.
[94] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, and A. Wu, "An Efficient k-Means Clustering Algorithm: Analysis and Implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.
[95] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2009.
[96] T. Niknam, E. Taherian Fard, N. Pourjafarian, and A. Rousta, "An efficient hybrid algorithm based on modified imperialist competitive algorithm and K-means for data clustering," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 2, pp. 306–317, Mar. 2011.
[97] A. McCallum, K. Nigam, and L. Ungar, "Efficient Clustering of High-Dimensional Data Sets with Application to Reference Matching," *Proceedings of the sixth ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, pp. 169–178, 2000.
[98] D. Irfan, X. Xiaofei, D. Shengchun, Z. He, and Y. Yunming, "S-Canopy: A feature-based clustering algorithm for supplier categorization," in *Industrial Electronics and Applications, 2009. ICIEA 2009. 4th IEEE Conference on*, 2009, pp. 677–681.
[99] C. Ranger, R. Raghuraman, A. Penmetsa, G. Bradski, and C. Kozyrakis, "Evaluating MapReduce for Multi-core and Multiprocessor Systems," in *IEEE 13th International Symposium on High Performance Computer Architecture, 2007. HPCA 2007*, 2007, pp. 13–24.
[100] M. Palmer, "Principal Components Analysis." [Online]. Available: http://ordination.okstate.edu/PCA.htm. [Accessed: 07-May-2013].
[101] "Linear Discriminant Analysis." [Online]. Available: http://www.lsv.uni-saarland.de/dsp_ss05_chap11.pdf. [Accessed: 07-May-2013].
[102] "Dendrogram plot - MATLAB." [Online]. Available: http://www.mathworks.co.uk/help/toolbox/stats/dendrogram.html. [Accessed: 14-Mar-2012].
[103] "How Dendrogram works." [Online]. Available: http://edndoc.esri.com/arcobjects/9.2/net/shared/geoprocessing/spatial_analyst_tools/how_dendrogram_works.htm. [Accessed: 14-Mar-2012].
[104] "Butterworth filter design - MATLAB." [Online]. Available: http://www.mathworks.co.uk/help/toolbox/signal/ref/butter.html. [Accessed: 20-Feb-2012].

[105] Texas Instruments, "TI active vs passive filters - Google Search," 2010. [Online]. Available: http://www.ti.com/lit/an/snoa224a/snoa224a.pdf. [Accessed: 20-Feb-2012].
[106] Microchip, "anti-aliasing, analog filters for data acquisition systems - Google Search," 1999. [Online]. Available: http://jimfranklin.info/microchipdatasheets/00699b.pdf. [Accessed: 20-Feb-2012].
[107] H. Akagi, "Modern active filters and traditional passive filters," *TECHNICAL SCIENCES*, vol. 54, no. 3, 2006.
[108] "Low Pass Butterworth Filter Design." [Online]. Available: http://www.electronics-tutorials.ws/filter/filter_8.html. [Accessed: 25-May-2012].
[109] Microchip, "dsPIC33FJ128GP206A," 2012. [Online]. Available: http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en541289. [Accessed: 01-Nov-2012].

Appendix A – DataCapture PCIBase II

```
#include <iostream>
#include <fstream>
#include <libad.h>
#include <conio.h>
#include "fft.h"
#include "calc.h"

using namespace std;

int main()
{
	int32_t adh;
	adh = ad_open("pcibase");
	if (adh==-1)
	{
		cout << "Failed to open PCIBaseII" << endl;
		return 0;
	}
	else
	{
		cout << "PCIBase Selected" << endl;
	}

	//Open Files for Analysis
	ofstream fV ("V.dat");
	ofstream fIT ("IT.dat");
	ofstream fI1 ("I1.dat");
	ofstream fI2 ("I2.dat");
	ofstream fI3 ("I3.dat");

	int32_t rc;
	int32_t scan_result;
	uint32_t run_id;
	struct ad_scan_cha_desc chav[5];
	struct ad_scan_desc sd;
	struct ad_scan_state state;

	memset(&chav, 0, sizeof(chav));
	memset(&sd, 0, sizeof(sd));

	chav[0].cha = AD_CHA_TYPE_ANALOG_IN|1;
	chav[0].store = AD_STORE_DISCRETE;
	chav[0].ratio = 1;
	chav[0].trg_mode = AD_TRG_NONE;
	chav[0].range = 0;

	chav[1].cha = AD_CHA_TYPE_ANALOG_IN|2;
	chav[1].store = AD_STORE_DISCRETE;
	chav[1].ratio = 1;
	chav[1].trg_mode = AD_TRG_NONE;
	chav[1].range = 0;

	chav[2].cha = AD_CHA_TYPE_ANALOG_IN|3;
	chav[2].store = AD_STORE_DISCRETE;
	chav[2].ratio = 1;
	chav[2].trg_mode = AD_TRG_NONE;
	chav[2].range = 0;

	chav[3].cha = AD_CHA_TYPE_ANALOG_IN|4;
	chav[3].store = AD_STORE_DISCRETE;
```

```
	chav[3].ratio = 1;
	chav[3].trg_mode = AD_TRG_NONE;
	chav[3].range = 0;

	chav[4].cha = AD_CHA_TYPE_ANALOG_IN|5;
	chav[4].store = AD_STORE_DISCRETE;
	chav[4].ratio = 1;
	chav[4].trg_mode = AD_TRG_NONE;
	chav[4].range = 0;



	float samples[20000];
	for (int i=0; i<20000; i++)
		samples[i]=0;

	sd.sample_rate = 0.00025f;
	sd.prehist = 0;
	sd.posthist = 22118400e58;
	sd.bytes_per_run = 4000;
	sd.ticks_per_run = 4000;

	rc = ad_start_scan(adh, &sd, 5, chav);
	if (rc<0)
	{
		cout << rc;
		return rc;
	}

	state.flags = AD_SF_SCANNING;

	while(state.flags & AD_SF_SCANNING)
	{
		rc = ad_get_next_run_f(adh, &state, &run_id, samples);
	//Debug Here
		if(rc!=0)
		{
			cout << rc;
			return rc;
		}

		cout << state.flags << ", " << state.runs_pending << ", " << run_id
<< endl;

		if(state.flags & AD_SF_SCANNING)
		{

		//Fill real and imaginary values from the read samples
		for (int i=0; i<4000; i++)
		{
			fV << samples[i]*894 << ", ";
			fIT << samples[i+4000]*36.6569 << " ,";
			fI1 << samples[i+8000]*36.6569 << " ,";
			fI2 << samples[i+12000]*36.5497 << " ,";
			fI3 << samples[i+16000]*36.6569 << " ,";
		}
		fV << endl;
		fIT << endl;
		fI1 << endl;
		fI2 << endl;
		fI3 << endl;

		for (int i=0; i<20000; i++)
			samples[i]=0;
```

```
            }
            if(kbhit())
            {
                  cin.get();
                  break;
            }
      }

      rc = ad_stop_scan(adh, &scan_result);
      cout << scan_result;
      ad_close(adh);

      return 0;
}
```

Appendix B – Load Operating Conditions Code

```
function GroupDefine=GroupDefine(count, cent, bin)

% The function filters throught the results from the histogram
function,
% and determines where there are groups of data that are available for
the
% classification of groups. This will be completed using the test
variable
% to monitor where the start of a group begins (i.e. test = 1
indicates the
% filter for the start of a group). Once the start of a group has been
% found then there is a function that will search fot the end of the
group
% (i.e test=2)

%To begin a group there needs to be more than 3 variables within the
bin,
%any less than this will be considered to be a transitional point and
will
%not be needed for the grouping of the data. Should the singular
datapoint
%be followed by a bin containing more than 3 datapoints, then this may
be
%considered as the start of a group

%There is also the issue around the harmonic not making an impact
(purly
%resistive load and there is just noise in the harmonics), any noise
should
%be considered as <0.001A)
test = 1;
i=1;
j=1;
s=size(count,1);
% Carry out the test until there are no more
while(i<=s)
       if (test==1) %Searching for the begining of a group
           if(i~=s)
                if(count(i)>=3||count(i)>0&&count(i+1)>=3)
                    GroupDefine(j,1)=cent(i)-bin;
                    test=2;
                end
           elseif (i==s)
               if(count(i)>=3)
                    GroupDefine(j,1)=cent(i)-bin;
                    GroupDefine(j,2)=cent(i)+bin;
                    break;
               else
                    break;
               end
           end
           if(i==s)
               break;
           end
           i=i+1;
       elseif (test==2) %Search for end of group
```

```
                if(count(i)==0||i==s)    %Allocate end of group if
found or end of dataset(i=s)
                    GroupDefine(j,2)=cent(i)+bin;
                    test=1;
                    j=j+1;
                    if(i==s)
                        break;  %break if i=s (End of Group)
                    end
                    i=i+1;
                else    %Still in the group therefore check next bin
                    i=i+1;
                end

        end
end

if (GroupDefine(size(GroupDefine, 1),2)-GroupDefine(1,1)<0.001)
    GroupDefine=[0,100];
End


function GroupSort=GroupSort(data, GroupFund, GroupHarmon)

for i=1:1:size(data, 1)
    for j=1:1:size(GroupFund, 1)
        if data(i,1)>=GroupFund(j,1) && data(i,1)<=GroupFund(j,2)
            GroupSort(i,1)=j;
            break;
        else
            GroupSort(i,1)=0;
        end
    end
    for j=1:1:size(GroupHarmon, 1)
        if data(i,2)>=GroupHarmon(j,1) && data(i,2)<=GroupHarmon(j,2)
            GroupSort(i,2)=j;
            break;
        else
            GroupSort(i,2)=0;
        end
    end

    if GroupSort(i,1)==0||GroupSort(i,2)==0
        GroupSort(i,3)=0;
    else
        GroupSort(i,3)=(GroupSort(i,1)*10+GroupSort(i,2));
    end

end
```

Appendix C – Data Import MATLAB

```
function [ITData, Analysis_Info]=CanopyAnalysisT1
%% Import Data into Matlab and
% Compute Harmonic Data, Angle and Magnitude of Data

%Filter Co-efficients
b=[0.0317 0.0951 0.0951 0.0317];
a=[1 -1.459 0.9104 -0.1978];

%Import and Filter
I1=importdata('C:\Users\dcarr\Documents\Thesis
Writing\Data\Results\LampHairDryer\I1.dat');
s=size(I1, 1);
I1=filtfilt(b,a,I1);
I1abs =(abs(fft(I1')')./(2000* sqrt(2)));
I1ang =(angle(fft(I1')'));
I1abs = I1abs(1:s,1:2001);
I2=importdata('C:\Users\dcarr\Documents\Thesis
Writing\Data\Results\LampHairDryer\I2.dat');
I1=filtfilt(b,a,I1);
I2abs = (abs(fft(I2')')./(2000* sqrt(2)));
I2ang =(angle(fft(I2')'));
I2abs = I2abs(1:s,1:2001);
IT=importdata('C:\Users\dcarr\Documents\Thesis
Writing\Data\Results\LampHairDryer\IT.dat');
IT=filtfilt(b,a,IT);
ITabs = (abs(fft(IT')')./(2000* sqrt(2)));
ITang =(angle(fft(IT')'));
ITabs = ITabs(1:s,1:2001);
V=importdata('C:\Users\dcarr\My Dropbox\Documents\Thesis
Writing\Data\Results\LampHairDryer\V.dat');
V=filtfilt(b,a,V);
Vabs = (abs(fft(V')')./(2000* sqrt(2)));
Vang =(angle(fft(V')'));
Vabs = Vabs(1:s,1:2001);

%% Create Dataset Arrays for Currents & Voltage

I1d=dataset({I1abs(:,1), 'DC'}, {I1abs(:,51), 'FundHarmon'},
{I1abs(:,101), 'Harmon1'},{I1abs(:,151), 'Harmon2'}, {I1abs(:,201),
'Harmon3'}, {I1abs(:,251), 'Harmon4'},{I1abs(:,301), 'Harmon5'},
{I1abs(:,351), 'Harmon6'}, {I1abs(:,401), 'Harmon7'}, {I1abs(:,451),
'Harmon8'}, {I1abs(:,501), 'Harmon9'});
%clear I1 I1abs
I2d=dataset({I2abs(:,1), 'DC'}, {I2abs(:,51), 'FundHarmon'},
{I2abs(:,101), 'Harmon1'},{I2abs(:,151), 'Harmon2'}, {I2abs(:,201),
'Harmon3'}, {I2abs(:,251), 'Harmon4'},{I2abs(:,301), 'Harmon5'},
{I2abs(:,351), 'Harmon6'}, {I2abs(:,401), 'Harmon7'}, {I2abs(:,451),
'Harmon8'}, {I2abs(:,501), 'Harmon9'});
%clear I2 I2abs
ITd=dataset({ITabs(:,1), 'DC'}, {ITabs(:,51), 'FundHarmon'},
{ITabs(:,101), 'Harmon1'},{ITabs(:,151), 'Harmon2'}, {ITabs(:,201),
'Harmon3'}, {ITabs(:,251), 'Harmon4'},{ITabs(:,301), 'Harmon5'},
{ITabs(:,351), 'Harmon6'}, {ITabs(:,401), 'Harmon7'}, {ITabs(:,451),
'Harmon8'}, {ITabs(:,501), 'Harmon9'});
%clear IT ITabs
Vd=dataset({Vabs(:,1), 'DC'}, {Vabs(:,51), 'FundHarmon'},
{Vabs(:,101), 'Harmon1'},{Vabs(:,151), 'Harmon2'}, {Vabs(:,201),
'Harmon3'}, {Vabs(:,251), 'Harmon4'},{Vabs(:,301), 'Harmon5'},
```

```
{Vabs(:,351), 'Harmon6'}, {Vabs(:,401), 'Harmon7'}, {Vabs(:,451),
'Harmon8'}, {Vabs(:,501), 'Harmon9'});
%clear V Vabs

%% Determine the operating conditions of the Loads
% This part will look at the individual loads and determine the groups
% within the load that show the different operating conditions of the
loads
% as these will influence the overall clustering process

% Using the hist functions the fundamental and harmonic can be grouped
into
% bins

% Load 1
[Fund1count, Fund1cent] = hist(I1d.FundHarmon, 60);
[Harmon1count, Harmon1cent] = hist(I1d.Harmon2, 60);

% Load 2
[Fund2count, Fund2cent] = hist(I2d.FundHarmon, 60);
[Harmon2count, Harmon2cent] = hist(I2d.Harmon2, 60);

% Create Dataset
LoadGroup=dataset({Fund1count', 'Fund1count'}, {Fund1cent',
'Fund1cent'},{Harmon1count', 'Harmon1count'}, {Harmon1cent',
'Harmon1cent'},{Fund2count', 'Fund2count'}, {Fund2cent',
'Fund2cent'},{Harmon2count', 'Harmon2count'}, {Harmon2cent',
'Harmon2cent'});
clear Fund1count Fund1cent Harmon1cent Harmon1count Fund2count
Fund2cent Harmon2cent Harmon2count

%Calculate the distance between centers and bin edge
bin1Fund = (LoadGroup.Fund1cent(2) - LoadGroup.Fund1cent(1))/2;
bin1Harmon = (LoadGroup.Harmon1cent(2) - LoadGroup.Harmon1cent(1))/2;
bin2Fund = (LoadGroup.Fund2cent(2) - LoadGroup.Fund2cent(1))/2;
bin2Harmon = (LoadGroup.Harmon2cent(2) - LoadGroup.Harmon2cent(1))/2;

BinSize = dataset({bin1Fund, 'Fund1'}, {bin1Harmon, 'Harmon1'},
{bin2Fund, 'Fund2'}, {bin2Harmon, 'Harmon2'});

% Need to calculate the groupboundaries for all of the fundamental and
% harmonic components
group1Fund=GroupDefine(LoadGroup.Fund1count, LoadGroup.Fund1cent,
bin1Fund);
group1Harmon=GroupDefine(LoadGroup.Harmon1count,
LoadGroup.Harmon1cent, bin1Harmon);
if(max(I1d.Harmon2)/max(I1d.FundHarmon)*100<3)
    clear group1Harmon;
    group1Harmon(1,1)=0;
    group1Harmon(1,2)=max(I1d.Harmon2);
end
group2Fund=GroupDefine(LoadGroup.Fund2count, LoadGroup.Fund2cent,
bin2Fund);
group2Harmon=GroupDefine(LoadGroup.Harmon2count,
LoadGroup.Harmon2cent, bin2Harmon);
if(max(I2d.Harmon2)/max(I2d.FundHarmon)*100<3)
    clear group2Harmon;
    group2Harmon(1,1)=0;
    group2Harmon(1,2)=max(I2d.Harmon2);
end
```

```
Group1=GroupSort([I1d.FundHarmon, I1d.Harmon2], group1Fund,
group1Harmon);
Group2=GroupSort([I2d.FundHarmon, I2d.Harmon2], group2Fund,
group2Harmon);
%Calculate Total Group Allocation
GroupTotal=(Group1(:,3).*100+Group2(:,3));
% Exclude all Transition Points
for i=1:1:length(GroupTotal)
    if(Group1(i,3)==0||Group2(i,3)==0)
        GroupTotal(i)=0;
    end
end

%% Monitoring Groups for further calculations

% The following will cycle through the group totals and determine how
many
% individual groups there are and store the group names in a seperate
% variable.

% The unique function returns all unique variables within the array
group=unique(GroupTotal);

%% Determine the range and avergae of all the groups

s=length(group);
Analysis=zeros(s,2);
for i=1:1:length(GroupTotal)
    for j=1:1:s
        if(GroupTotal(i)==group(j))
            Analysis(j,1)=Analysis(j,1)+1;
            Analysis(j,2)=Analysis(j,2)+ITd.FundHarmon(i);
        end
    end
end
ITd.group=GroupTotal;
% Find Max and Min values for the fundamantal harmonics
% Fill the max and min variable with values for use in comparison, 50
has
% been chosen as this will be greater than the maximum current
available

for i=1:1:s
    Analysis(i,3)=0;
    Analysis(i,4)=50;
end

% Cycle through the fundamental values for total current draw and
check
% group and max/min of value

for i=1:1:length(GroupTotal)

    for j=1:1:s
        if GroupTotal(i)==group(j)&&GroupTotal(i)~=0
            if ITd.FundHarmon(i) > Analysis(j,3)
                Analysis(j,3)=ITd.FundHarmon(i);
            end
            if ITd.FundHarmon(i,1) < Analysis(j,4)
                Analysis(j,4)=ITd.FundHarmon(i);
            end
```

```
        end
    end
end

% Calculate Range And Average of groups

Analysis(:,5)=Analysis(:,2)./Analysis(:,1);
Analysis(:,6)=Analysis(:,3)-Analysis(:,4);

% Calculate Ratio of Center to radii (Half Range)

Analysis(:,7)=(Analysis(:,6)./2)./Analysis(:,5);

AnalysisInfo=dataset({group, 'Group'}, {Analysis(:,1), 'Count'},
{Analysis(:,2), 'Total'}, {Analysis(:,3), 'Min'}, {Analysis(:,4),
'Max'}, {Analysis(:,5), 'Average'}, {Analysis(:,6), 'Range'},
{Analysis(:,7), 'Ratio'});
clear Analysis;

ITData=single(ITd);
Analysis_Info=single(AnalysisInfo);
```

Appendix D – Canopy Clustering Algorithm

```
%% Mapper Function
% Splits the data into two data sets for canop clustering to be
completed
% on each of the individual sets
i=length(ITd.FundHarmon);
if(mod(i,2))
    temp=i+1;
else
    temp=i;
end

mapping=randperm(i);
mapper1=mapping(1:temp/2);
mapper2=mapping((temp/2)+1:i);

for temp=1:1:length(mapper1)
    x1(temp)=ITd.FundHarmon(mapper1(temp));
    y1(temp)=ITd.Harmon2(mapper1(temp));
    g1(temp)=ITd.group(mapper1(temp));
end
for temp=1:1:length(mapper2)
    x2(temp)=ITd.FundHarmon(mapper2(temp));
    y2(temp)=ITd.Harmon2(mapper2(temp));
    g2(temp)=ITd.group(mapper2(temp));
end
clear temp;

%% Canopy Cluster Mapper Function
% Computes the centres for the canopies within the mappers only

% Constants for the ratio function for the t1 boundary
a1=0.00016;
b1=-0.0041;
c1=0.029;

% Mapper 1
j=1;
temp=1;
centertemp1=ones(size(x1));
    for count=1:1:length(x1)
        if (centertemp1(count)==1)     %Can be used as canopy center
            centertemp1(count)=2;      %Used for Canopy Center
            center1(temp)=mapper1(count);
            center(j,1)=ITd.FundHarmon(mapper1(count));
            center(j,2)=ITd.Harmon2(mapper1(count));
            j=j+1;
            temp=temp+1;
            t1=(x1(count)*(a1*(x1(count)^2)+b1*x1(count)+c1));
            for count1=1:1:length(x1)
                if(centertemp1(count1)==1&&count1~=count)
                    calc=sqrt(((x1(count)-x1(count1))^2)+((y1(count)-
y1(count1))^2));
                    if calc<=t1;  %Within the radius (half of the
range)
                        centertemp1(count1)=0;     %Cannot be used as
canopy center
                    end
                end
```

```
            end
        end
    end
gscatter(x1, y1, g1)
hold
for count=1:1:length(centertemp1)
    if (centertemp1(count)==2)
        plot(x1(count), y1(count), 'bx', 'markersize',30)
        circle([x1(count),
y1(count)],(x1(count)*(a1*(x1(count)^2)+b1*x1(count)+c1)), 1000,
'r:');

%circle([handles.current_data(handles.centers(count),handles.x),

%handles.current_data(handles.centers(count),handles.y)],handles.curre
nt_data(handles.centers(count),handles.x)*handles.t2, 1000, 'b-');
    end
end

% Mapper 2
centertemp2=ones(size(x2));
    for count=1:1:length(x2)
        if (centertemp2(count)==1)     %Can be used as canopy center
            centertemp2(count)=2;      %Used for Canopy Center
            center1(temp)=mapper2(count);
            center(j,1)=ITd.FundHarmon(mapper2(count));
            center(j,2)=ITd.Harmon2(mapper2(count));
            j=j+1;
            temp=temp+1;
            t1=(x2(count)*(a1*(x1(count)^2)+b1*x1(count)+c1));
            for count1=1:1:length(x2)
                if(centertemp2(count1)==1&&count1~=count)
                    calc=sqrt(((x2(count)-x2(count1))^2)+((y2(count)-
y2(count1))^2));
                    if calc<=t1;  %Within the radius (half of the
range)
                        centertemp2(count1)=0;     %Cannot be used as
canopy center
                    end
                end
            end
        end
    end
gscatter(x2, y2, g2)
hold
for count=1:1:length(centertemp2)
    if (centertemp2(count)==2)
        plot(x2(count), y2(count), 'bx', 'markersize',30)
        circle([x2(count),
y2(count)],(x2(count)*(a1*(x2(count)^2)+b1*x2(count)+c1)), 1000,
'r:');

%circle([handles.current_data(handles.centers(count),handles.x),

%handles.current_data(handles.centers(count),handles.y)],handles.curre
nt_data(handles.centers(count),handles.x)*handles.t2, 1000, 'b-');
    end
end

% Map Reduce
clear centerFinal;
j=1;
centertemp=ones(length(center),1);
```

```
for count=1:1:length(centertemp)
        if (centertemp(count)==1)     %Can be used as canopy center
            centertemp(count)=2;      %Used for Canopy Center
            xtemp=center(count,1);
            ytemp=center(count,2);
            monitor=1;

t1=(center(count,1)*(a1*(center(count,1)^2)+b1*center(count,1)+c1));
            for count1=1:1:length(center)
                temp=sqrt(((center(count,1)-
center(count1,1)).^2)+((center(count,2)-center(count1,2)).^2));
                if temp<=t1;  %Within the radius (half of the range)
                    centertemp(count1)=0;     %Cannot be used as canopy
center
                    xtemp=xtemp+center(count1,1);
                    ytemp=ytemp+center(count1,2);
                    monitor=monitor+1;
                end
            end
            centerFinal(j,1)=(xtemp/monitor);
            centerFinal(j,2)=(ytemp/monitor);

t1Final(j,1)=(centerFinal(j,1)*(a1*(centerFinal(j,1)^2)+b1*centerFinal
(j,1)+c1))
            j=j+1;
        end
end

gscatter(ITd.FundHarmon, ITd.Harmon2, ITd.group)
hold
for count=1:1:length(centerFinal)
        plot(centerFinal(count,1), centerFinal(count,2), 'bx',
'markersize',30)
        circle([centerFinal(count,1),
centerFinal(count,2)],t1Final(count,1), 1000, 'r:');
end

for i=1:1:length(ITd.FundHarmon)
    for j=1:1:length(centerFinal)
        distance(i,j)=sqrt(((ITd.FundHarmon(i)-
centerFinal(j,1)).^2)+((ITd.Harmon2(i)-centerFinal(j,2)).^2));
        if distance(i,j)<=t1Final(j,1);
            distance(i,j)=1;
        else
            distance(i,j)=0;
        end
    end
end

can=1;
mem=1;
for i=1:1:length(ITd.FundHarmon)
    for j=1:1:length(centerFinal)
        if distance(i,j)==1
            CanopyMembership(mem, can)=j;
            can=can+1;
        end
    end
    can=1;
    mem=mem+1;
end
```

```
CanopyMembership=unique(CanopyMembership, 'rows');

%% k-means clustering witrhin the overlapping canopies
% This stage is comepleted after removing the duplicates within the
% Canop[yMonitoring variable, and is now filtered through to give the
k
% centers and the allocation of the cluster to each of the points
kOut=zeros([size(CanopyMembership,1), 2]);
kMonitor=0;

for i=1:1:size(CanopyMembership, 1)
    for j=1:1:size(CanopyMembership, 2)
        for k=1:1:size(distance,1)
            if CanopyMembership(i,j)>0
                if distance(k,(CanopyMembership(i,j)))==1
                    kOut(i,1)=kOut(i,1)+ITd.FundHarmon(k);
                    kOut(i,2)=kOut(i,2)+ITd.Harmon2(k);
                    count=count+1;
                    kMonitor(k,1)=i;
                end
            end
        end
    end
    kOut(i,1)=kOut(i,1)./count;
    kOut(i,2)=kOut(i,2)./count;
    count=0;
end

%% Load Combinations
% Find the different load combintations and calculate the phase angles

IT1ang=Vang-I1ang;
IT2ang=Vang-I2ang;
ITTang=Vang-ITang;

alpha=abs(IT1ang(:,51)-IT2ang(:,51));
c=sqrt(I1d.FundHarmon.^2+I2d.FundHarmon.^2.+(2.*I1d.FundHarmon.*I2d.Fu
ndHarmon.*alpha));
alpha3=abs(IT1ang(:,151)-IT2ang(:,151));
c3=sqrt(I1d.Harmon2.^2+I2d.Harmon2.^2.+(2.*I1d.Harmon2.*I2d.Harmon2.*c
os(alpha3)));
```